COMMODORE 64

HANDBOEK



Commodore

WAARSCHUWING

Dit toestel wekt HF straling op die bij onjuiste verbinding met de TV ontvanger storing kan veroorzaken in andere radio- en TV-toestellen. Volg nauwkeurig de aanwijzingen op welke zijn vermeld op pagina 3 en 4 om dit risico zoveel mogelijk te vermijden. In het geval dat er toch storing optreedt bij TV- of radio-ontvangst, test dan eerst of die storing afkomstig is van uw Commodore 64. (Zet uw computer even aan en weer uit).
Indien ja, probeer dan het volgende:

- draai aan de antenne;
- zet de computer verder van het ontvangtoestel af;
- is het ontvangtoestel correct verbonden met de antenne of CAI systeem?

In het geval dat bovenstaande raadgevingen geen verbetering geven, kunt u de CBM servicedienst van uw dealer bellen.

HANDBOEK VOOR DE COMMODORE 64

Uitgegeven door
Commodore Business Machines Inc.

Vertaling: Commodore Nederland

tweede druk : 840264-C1

Inhoudsopgave

Voorwoord 1

HOOFDSTUK 1

Installatie 3
Uitpakken van uw Commodore 64 4
Aansluiten op de Televisie en het lichtnet 5
Aansluiten op de Commodore Monitor 7
Aansluiten op andere Monitoren en versterkers 12
Check-List 13
Het aansluiten van de 1530 cassetterecorder 14
Instellen van de kleuren 16
Accessoires 18

HOOFDSTUK 2

Het grote begin 21
Toetsenbord 22
Het laden en saven van programma's 24
PRINT en rekenen 28
Rekenkundige bewerkingen 29
Meerdere berekeningen op een regel 30
Rekenvolgorde 31
Het afdrukken van tekst en getallen 32

HOOFDSTUK 3

Programmeren in Basic: Het grote begin 33
De volgende stap 34
GOTO en regelnummers 35
Het LIST commando 35
Tips voor het maken van veranderingen 35
Variabelen 36
IF...THEN 38
FOR...NEXT loops 39

HOOFDSTUK 4

Basic voor gevorderden 41
Inleiding 42
Bewegende beelden 43
Loops of lussen 44
INPUT 45
GET 46
Functie toetsen 47
Random getallen en andere functies 48
Een raadseltje 50
Werpt U maar 51
Random graphics 52
CHR$ en ASC functies 52

HOOFDSTUK 5

Kleur en graphics voor gevorderden 53
Kleur en graphics 54
Het afdrukken van kleuren 54
CHR$ codes behorende bij de kleuren 56
PEEK en POKE 57
Tekenen op het scherm 59
De geheugenplaatsen van het scherm 59
De geheugenplaatsen voor de kleur informatie 60
Nog meer stuiterende ballen 61

HOOFDSTUK 6

Sprite graphics 63
Van bits en bytes naar sprites 64
Het tweetallig rekenstelsel 64
Het maken van sprites 67
Meer over sprites, kleur en beweging 73

HOOFDSTUK 7

Het maken van geluid 75
U bent geen programmeur, en toch wilt u geluid 76
De opbouw van een toon 76
Volume 77
Golfvorm 77
Frequentie 79
Omhullende generator (ADSR) 80

Attack/Decay 80
Sustain/Release 82
Voorbeeld programma's 83
Een melodie spelen met uw Commodore 64 85
Geluidseffecten 86
Het proberen waard 87

HOOFDSTUK 8

Data manipulatie voor gevorderden 88
READ en DATA 89
Middelen van getallen 91
Geïndiceerde variabelen 93
Het begrip dimensie 95
Dobbelstenen met array's 96
Twee-dimensionale array's 97
Enquete 98

APPENDICES

Voorwoord 102
A Beschikbare software 103
B gebruiksaanwijzing voor uw 1530 datasette 109
C Commodore 64 BASIC 120
D Afkortingen van BASIC woorden 134
E Scherm codes 136
F ASCII en CHR$ codes 139
G Memory map van het scherm- en kleurgeheugen 142
H Afgeleide wiskundige functies 144
I Aansluitgegevens 145
J Het proberen waard 148
K Het omzetten van programma's naar Commodore BASIC 151
L fout boodschappen 152
M Muzieknoten 154
N Literatuur opgave 158
O Sprite registers 161
P Muziek registers 164

INDEX 168

Voorwoord

Van harte gelukgewenst met de aankoop van een van de beste computers ter wereld! U bent de trotse bezitter van een Commodore 64. De Commodore 64 staat bekend als de vriendelijke computer en een deel van die vriendelijkheid vindt u terug in ons streven om te komen tot handboeken die gemakkelijk te lezen zijn, eenvoudig in het gebruik en: begrijpelijk. Dit handboek in het bijzonder is geschreven om u alle informatie te verschaffen voor het in bedrijfstellen van uw Commodore 64, en u op vlotte wijze op dreef te brengen bij het zelf maken van programma's.

Als u behoort tot de grote groep gebruikers die niet zelf willen programmeren: alle informatie om Commodore programma's op cassette of in cartridges te kunnen gebruiken, vindt u vooraan in het handboek. U hoeft dus niet alles door te bladeren om te kunnen beginnen!

Er zitten nogal wat buitengewone mogelijkheden in uw Commodore 64. In de eerste plaats bezit u wat grafische mogelijkheden aangaat, de meest geavanceerde 'plaatjesmaker' uit de hele microcomputer industrie. We noemen het "sprite graphics", en u bent in staat afbeeldingen te maken in 4 kleuren, precies zoals u dat ziet in de mooiste video speelautomaten. Bovendien kunt u 8 verschillende afbeeldingen tegelijkertijd over het scherm laten bewegen. U kunt voorwerpen over elkaar, achter elkaar langs laten gaan. De computer geeft aan of ze in botsing komen zodat u kunt bepalen wat er zal gaan gebeuren als uw creaties elkaar raken. De Commodore 64 heeft als tweede belangrijke eigenschap de ingebouwde muziek synthesizer die niet onderdoet voor vele reeds bekende merken. U kunt 3-stemmig muziek maken over een bereik van 9 octaven, 4 verschillende golfvormen: zaagtand, driehoeksspanning, blokgolf generator en ruis. Aanzwel- en afvaltijd, de duur en het nagalmvolume van het geluid is te programmeren. Standaard is ook de omhullende generator en de programmeerbare eindfiltering (bandpass, low- en high pass!) voor de drie stemmen samen. Natuurlijk volume instelling en zwevings modulator. De Commodore 64 kan aangesloten worden aan elke hi-fi installatie zodat uw muziekstukken professionele kwaliteit krijgen.

Randapparatuur

Nu we het toch hebben over het aansluiten van de Commodore 64 aan andere apparatuur: vergeet niet dat de computer uitgebreid kan worden met accessoires (randapparaten genoemd) wanneer u daar de behoefte toe voelt. Een voorbeeld daarvan is een cassetterecorder (de Datasette©) of een tot vier floppy 1541 disk drives om de door u zelf gemaakte programma's op vast te leggen.
De vele Commodore printers, matrix-printers zoals de MPS-801 en MPS-802, een kleuren printer MCS-801, een margrietwiel printer de DPS-1101 en de printer plotter 1520 kunnen zo aangesloten worden. U kunt dan programma's afdrukken, fakturen drukken, of brieven schrijven. Een 1701 kleuren monitor voor een perfecte beeldkwaliteit. Zelfs spraak is mogelijk, wanneer u uw 64 met de 'speech- module' uitrust. Indien u een van de enthousiaste CP/M gebruikers bent: de Z-80 CP/M cartridge verandert uw Commodore 64 in een CP/M machine en geeft u toegang tot de grote verscheidenheid aan CP/M software.
Hardware alleen is niet het enige dat belangrijk is. Dit handboek zal u helpen computers te leren begrijpen. Er staat niet alles over in, maar waar nodig wordt er verwezen naar dieper gaande informatie over het behandelde onderwerp.

Wij van Commodore hopen echt dat u veel plezier zult beleven aan uw nieuwe aanschaf. Programmeren is echter iets dat men niet in een dag kan leren, en om een goede basis te verkrijgen, dient u in het begin langzaam en geduldig het handboek door te werken. U zult hier later veel steun aan hebben. Vul voor u begint de garantiekaart in en zend deze op naar Commodore. U ontvangt dan regelmatig aanvullende informatie over het gebruik van uw computer.
Veel genoegen!

Noot:

Tijdens het ter perse gaan van dit handboek waren er nog veel programma's voor de Commodore 64 in ontwikkeling. Raadpleeg uw wederverkoper voor de laatste stand van zaken of abonneer u op een van de tijdschriften van Commodore gebruikers:

- VCGN
- VIC nieuws
- Printout
- Commodore 64

HOOFDSTUK 1
INSTALLATIE

HOOFDSTUK 1
INSTALLATIE

- Uitpakken van uw Commodore 64
- Het aansluiten aan de TV
- Het aansluiten van uw Commodore 1701/2 monitor
- Aansluiten op andere monitoren of versterkers
- Het aansluiten van de 1530 cassette recorder
- Aanzetten
- Instellen van de kleuren
- De Commodore 64 en haar accessoires

Uitpakken en aansluiten

Wanneer u de hiernavolgende aanwijzingen nauwkeurig opvolgt, kunt u er zeker van zijn dat alle verbindingen naar de TV, de hi-fi installatie of naar een eventuele monitor in orde zijn en dat alles goed zal werken. Controleer om te beginnen de inhoud van de polystyreen doos. Daar behoort in te zitten:

1. 1 Commodore 64 computer
2. 1 voeding (een dik beige blok met twee snoeren eraan)
3. 1 coaxiale kabel voor het aansluiten van de TV
4. 1 manual (Nederlands)
5. 1 garantiekaart (Nederlands)

Indien een van de bovengenoemde punten ontbreekt, dient u direkt kontakt op te nemen met uw wederverkoper.
Laten we eerst eens bekijken wat er zoal aan schakelaars en aansluitpunten aanwezig zijn op uw computer.

Rechterzijkant

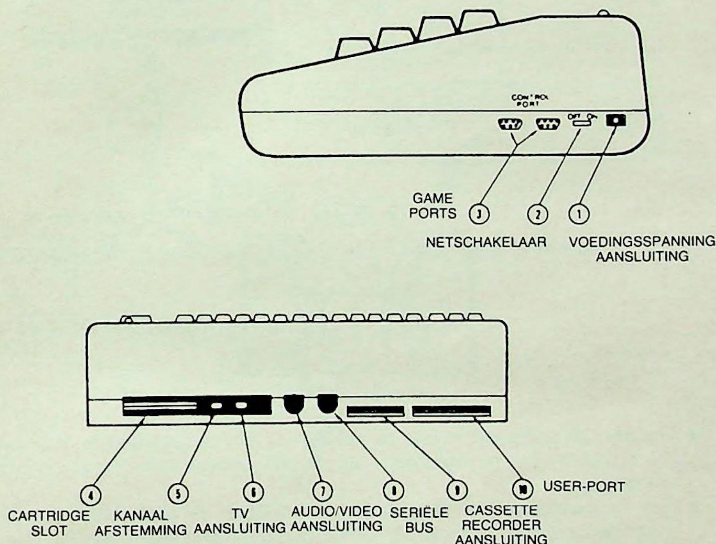
1. Aansluiting voor de voedingsspanning. Dit is het punt waar u het voedingsblok aan moet sluiten. De ronde plug aan een van de snoeren past hier precies in.
2. Netschakelaar. Hiermee schakelt u de computer aan en weer uit.
3. Aansluitingen voor joysticks of paddles voor spelen. Hier kan ook een lichtpen op worden aangesloten.

Achterzijde

4. Cartridge aansluiting. Deze rechthoekige connector is bestemd voor het insteken van cartridges met programma's of spelletjes.

5. Hoogfrequent uitgang. Via deze bus komt het signaal voor uw TV naar buiten (beeld en geluid).
6. Audio en video uitgang. Deze DIN-connector levert een hi-fi audio signaal voor aansluiting aan een geluidsinstallatie, alsmede een PAL- composiet video signaal bestemd voor een (kleuren) monitor.
7. Aansluiting voor randapparatuur, de z.g. seriële poort. Printers of disk drives worden alle aan deze poort aangesloten.
8. Aansluiting voor cassetterecorder. Hier kan de speciale Datasette recorder worden aangesloten voor het opslaan van informatie
9. De z.g. userport. Deze in- en uitgang is bestemd voor speciale doeleinden zoals MODEM aansluiting of RS 232 communicatie.

NB: RANDAPPARATUUR EN CARTRIDGES NOOIT AANSLUITEN OF VERWIJDEREN MET AANGESCHAKELDE COMPUTER!!!!



Aansluiten op de tv

1. Steek de smalle connector van de coaxiale kabel in connector 5. Goed aandrukken, het kan maar op één manier.
2. Steek het andere einde in de coaxiale antenne-ingang van uw TV.
3. Steek de ronde aansluitbus van het voedingsblok in connector 1, en steek de stekker in het stopcontact (220V, 50Hz). De 220 V van het lichtnet wordt omgevormd tot een lage spanning van 5V en 9V. De ronde aansluitbus kan maar op een manier aan de Commodore 64 worden aangesloten. Alle noodzakelijke verbindingen zijn gemaakt.

NAAR HF AANSLUITING

ACHTERKANT TV

UHF

VHF

ANTENNE INGANG

COMPUTER

TV

ANTENNE SCHAKELDOOS

VHF ANTENNE

AUDIO/VIDEO AANSLUITING

NAAR AUXILIARY-INGANG OF TUNER-INGANG

NAAR VIDEO IN

TV-MONITOR

VERSTERKER

4. Schakel de computer in met de schakelaar aan de rechterzijde. Het lampje op de computer zal nu oplichten. Als dit niet gebeurt kijk dan in de 'checklist' verderop.
5. De TV moet ingesteld worden op kanaal 36 in de UHF band. Kijk in de gebruiksaanwijzing van uw TV toestel hoe u dit moet doen. Stel de TV zodanig af dat u een beeld krijgt als volgt:

```
**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY.
```



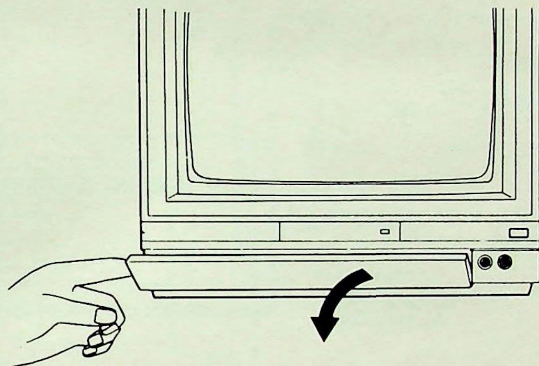
EEN KNIPPERENDE CURSOR GEEFT AAN DAT DECOMMODORE-64 OP UW INSTRUKTIE WACHT

De Commodore 1701/2 kleurenmonitor

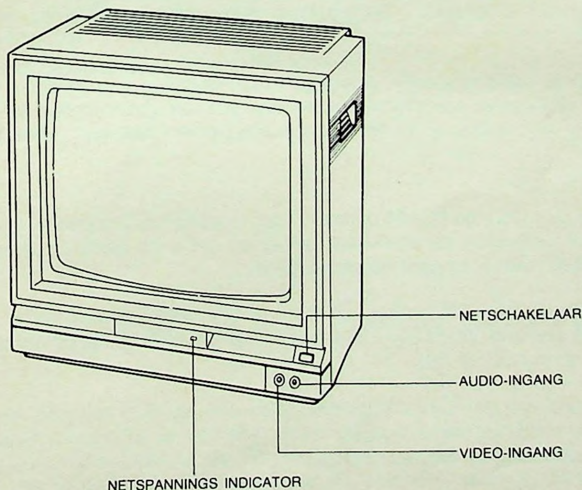
Als u een Commodore 1701/2 monitor heeft aangeschaft om een beter beeld te verkrijgen, dan vervalt de uitleg over het aansluiten op de TV en is het navolgende stukje uitleg van belang.

Benaming van de bedieningsknoppen

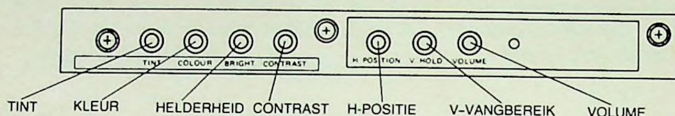
VOORZIJDE



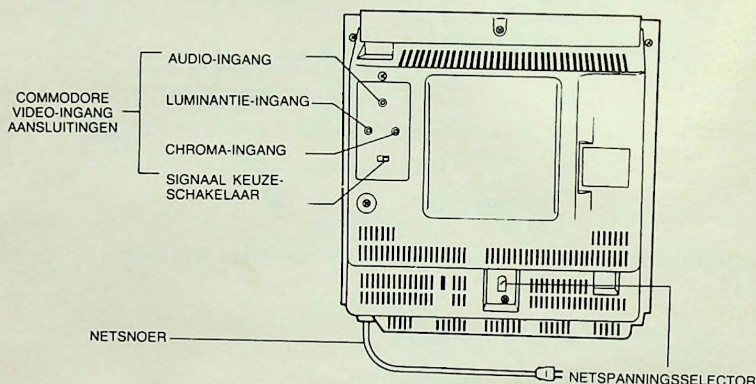
Deze klep kan opengeklapt worden door deze naar u toe te trekken.



BEDIENINGSKNOPPEN VOORKANT



ACHTERZIJDE

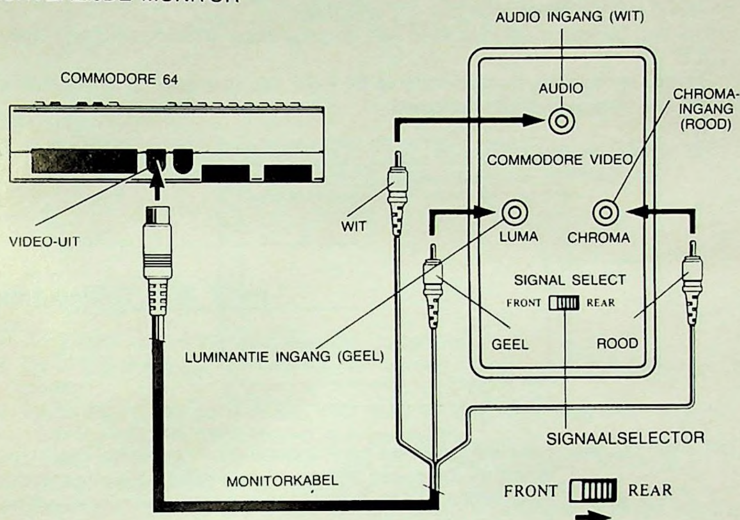


Om de monitor op uw Commodore 64 computer aan te sluiten maakt u gebruik van de bij de monitor geleverde aansluitkabel. Let er op dat u de beide apparaten uitgeschakeld heeft voor u ze met elkaar verbindt.

1. Steek de ronde 5-polige DIN-steker achter in uw Commodore 64 computer, in de aansluiting gemerkt VIDEO-UIT (Dit is de linker aansluiting, bij achteraanzicht van uw Commodore 64).

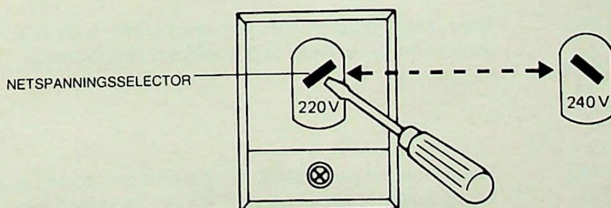
   Aan het uiteinde van de verbindingskabel zitten drie zg. RCA pluggen, deze moeten op de volgende wijze, aan de achterzijde van de monitor, gestoken worden : de audio-uit plug WIT naar AUDIO, luminantie-uit GEEL naar LUMA en de chroma-uit ROOD naar CHROMA. De schakelaar SIGNAL-SELECT moet in de stand 'REAR' (achter) geplaatst worden.

ACHTERZIJDE MONITOR

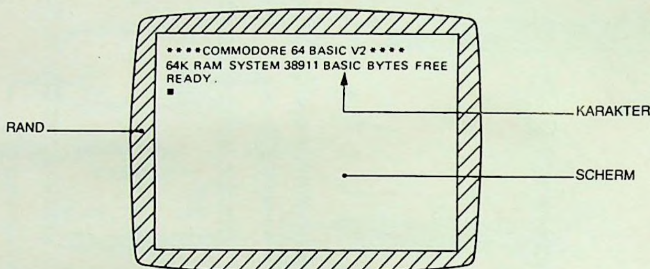


Het is mogelijk een VIDEO-RECORDER of TV-TUNER op uw 1701/2 monitor aan te sluiten, u moet dan gebruik maken van de aansluitingen aan de voorzijde van de monitor. De schakelaar 'SIGNAL SELECT' dient dan in de stand 'FRONT' te staan. Voor gedetailleerde informatie verwijzen wij u naar de bij de monitor geleverde handleiding.

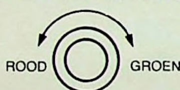
2. Controleer of de ingestelde netspanning inderdaad 220 V is. Dit kunt u zien op de netspannings selector aan de achterzijde van de monitor. Stel deze indien nodig, met een kleine schroevendraaier in op 220 V.



3. Schakel de monitor AAN door op de schakelaar aan de voorzijde te drukken. De groene netspannings indicator zal oplichten.
(Het een tweede maal indrukken van de schakelaar schakelt de monitor weer UIT).
Schakel vervolgens uw Commodore 64 AAN. Als alles goed is zal de monitor met het volgende beeld opkomen.

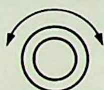


TINT INSTELLING Naar links draaien geeft meer rood, naar rechts geeft meer groen. Probeer een zo natuurlijk mogelijke kleur-verhouding in te stellen. De normale stand van deze knop is het midden (klik- positie).



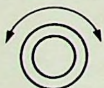
KLEUR INSTELLING

Draai naar links voor fletse (zwart/wit) kleuren en naar rechts voor volle warme kleuren. De normale stand van deze knop is het midden (klik- positie).



HELDERHEID INSTELLING

Naar rechts draaien voor een helder beeld. De normale stand van deze knop is het midden (klik- positie).



CONTRAST INSTELLING

Naar rechts draaien voor een contrastrijk beeld. De normale stand van deze knop is het midden (klik-positie).

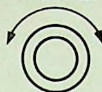


H.POSITIE INSTELLING

Draai de knop naar rechts, om het midden van het beeld naar rechts te verplaatsen, en naar links m het scherm naar links te verplaatsen. De normale stand van deze knop is het midden (klik-positie).

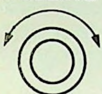


V.VANG BEREIK



Verdraai deze knop naar links of naar rechts om het verticale rollen van het beeld tegen te gaan.

VOLUME INSTELLING



Draai deze knop naar rechts om het geluids volume te versterken, en naar links om het te verminderen.

BELANGRIJKE TIPS

- Zorg voor voldoende ventilatie van de monitor.
- Bedek de bovenste ventilatie sleuven van de monitor nooit met papier, of een doek.
- Plaats de monitor nooit op een bed, sofa, of wollig tapijt, waardoor de onderste ventilatie sleuven geblokkeerd kunnen worden.
- Plaats de monitor nooit in een kast, als er niet voldoende ventilatie is.
- Gebruik geen water in de directe omgeving van de monitor.
- Overtuig u ervan dat kinderen geen metalen voorwerpen door de ventilatie openingen naar binnen steken.

Aansluiten op andere monitoren of versterkers

U kunt het geluid van de Commodore 64 weer laten geven via een hi-fi versterker installatie. Bovendien is er voorzien in een PAL-composiet video signaal, zodat weergave van beeld en geluid via een andere dan de Commodore 1701/2 monitor ook tot de mogelijkheden behoort.
Verbindingen naar die apparatuur dienen te lopen via de audio-video uitgang 6 aan de achterzijde. Om het u gemakkelijk te maken zijn er in de radiohandel kabels te koop met aan een zijde een 5-pens DIN audio aansluiting, en aan de andere een rode en een zwarte coaxiale plug. Goedkoper is zelf maken: de aansluitgegevens vindt u in appendix I.
De zwarte plug van de DIN kabel is de audio uitgang. Indien u toevallig een apparaat bezit met een audio ingang voor die plug, kunt u hem zo aansluiten. Gebruik de ingang gemerkt AUX voor dit doel.
De witte (of de rode) connector is de video uitgang welke naar de monitor gaat. (De 5-polige 180 gr plug past in de 8-polige aansluiting van de Commodore 64).
In appendix I staat aangegeven hoe de verbindingen horen te zijn in het geval u ondanks alle moeite nog geen beeld of geluid krijgt.De bezitters van een 1541 disk drive of een van de vele Commodore seriële printers, dienen hun toestel aan te sluiten volgens in de bijbehorende handboeken aangegeven instructies.

Resume

1. Zet de TV aan op UHF band
2. Zet de computer aan (netschakelaar 2 aan zijkant)
3. Na even wachten ziet u het volgende beeld:
4. Stel de fijnregeling van de kanaalkiezer zo in dat u een scherp beeld krijgt.
5. Instellen van kleur heeft nu nog niet veel zin. Het beeld is donkerblauw met lichtblauwe rand en letters.

Checklist

Volg onderstaande checklist indien u geen juist beeld heeft.

Symptoom	Oorzaak	Remedie
Lampje (power) niet aan.	Computer uit	Kijk of schakelaar wel "on" staat.
	Voedingskabel niet in connector 1	Voedingskabel goed aandrukken in connector.
	Voeding niet aangesloten	Steek stekker in stopcontact
	Zekering in computer doorgeslagen.	Ga naar de dealer voor vervanging en controle
Geen beeld, wel sneeuw	TV niet op juiste kanaal	Probeer UHF kanaal 36
	Computer niet aan TV aangesloten.	Aansluiten!
Slecht beeld	niet van toepassing	Draai aan kanaalkiezer
Onzin op beeld bij gebruik van een cartridge.	Cartridge verkeerd ingestoken.	Computer uitzetten, cartridge goed insteken, computer aanschakelen.
Beeld zonder kleur	Slechte afstemming	Goed afstemmen.
Slechte kleuren	Kleurafstelling TV onjuist	Contrast en kleurverzadiging instellen.
Goed beeld, sissend geluid.	Volume regelaar TV te ver open.	TV zachter zetten.
Beeld OK, geen geluid.	TV geluid te zacht. Aux input fout verbonden of niet geselecteerd.	TV harder zetten. Raadpleeg instructies van uw versterker.

TIP:
Iedereen kan werken met een Commodore 64 computer. Wij kunnen ons echter voorstellen dat er zich wel eens problemen voordoen. Commodore geeft tijdschriften en literatuur uit om veel van de vragen te beantwoorden en een schat aan voorbeelden te geven.

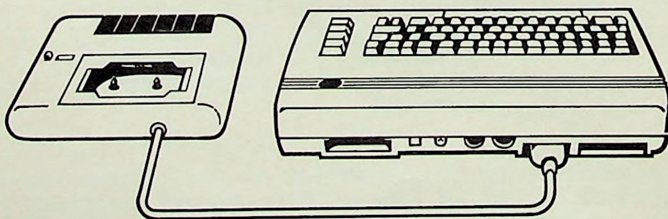
Het aansluiten van de 1530 cassetterecorder op uw Commodore 64
Voorlopige controle

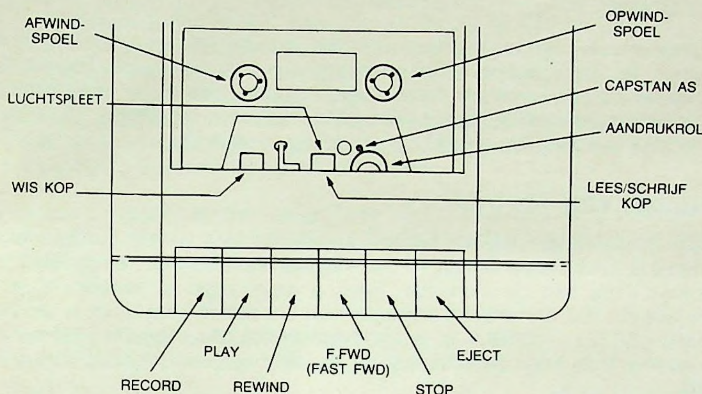
Voordat u uw 1530 gaat gebruiken om programma's in te lezen of op te slaan moet u eerst de werking controleren aan de hand van de volgende stappen:
1. Schakel de computer UIT, sluit de 1530 Cassette Recorder aan (zie figuur).

MAAK NOOIT EEN VERBINDING MET UW COMPUTER ALS DEZE IS AANGESCHAKELD

2. Verzeker u ervan dat de 1530 motor uit is door na te gaan of alle functietoetsen omhoog staan. Is dit niet het geval, dan drukt u op de STOP-toets.
3. Schakel de computer AAN.
4. Druk de PLAY-toets op de 1530 in. Kijk of, wanneer de toets wordt ingedrukt, de READ/WRITE-kop naar de spoelen toe beweegt en of de capstan-as in contact komt met de aandrukrol (zie figuur). De oprolspoel moet gelijkmatig tegen de klok indraaien.
5. Druk nu op de STOP-toets. De koppen moeten nu teruggaan en de spoel moet stoppen.
6. Druk de REWIND-toets in. De hoofden moeten in de rustpositie blijven en de aanvoerspoel moet snel met de wijzers van de klok meedraaien.
7. Druk nogmaals op STOP en vervolgens op F.FWD. De schrijfkoppen blijven nog steeds in de rustpositie en de oprolspoel moet nu snel tegen de wijzers van de klok indraaien.
8. Druk nogmaals op STOP en probeer dan VOORZICHTIG de RECORD-toets in te drukken. U moet hier een sterke mechanische weerstand gewaar worden. NIET doordrukken. Dit is een beveiliging.
9. Wanneer alles naar behoren heeft gewerkt, kunt u verder gaan met de controle van uw computer systeem. In hoofdstuk 2 zult u leren hoe u programma's kunt inlezen en wegschrijven. Wanneer u echter op problemen bent gestuit bij deze voorlopige controle, wilt u dan de laatste pagina van de cassette-recorder handleiding opslaan. Hier staan een aantal tips beschreven, waarmee u uw problemen kunt oplossen

CASSETTE RECORDER AANSLUITING





Controle op de werking

Om de werking van uw nieuwe 1530 te testen moet u een kort programma schrijven, en dit weg schrijven op cassette (SAVEn) om het hierna weer in te lezen (LOADen) in de computer. Zie hoofdstuk 2 en de cassette recorder handleiding.

De 1530 gebruikt normale audiocassettes. Druk altijd op REWIND om ervoor te zorgen dat u aan het begin van het bandje bent.

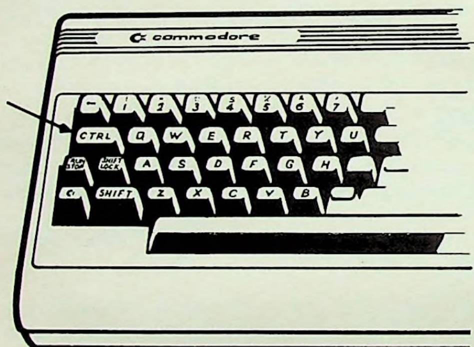
Noot: Gebruik bandjes met een speeltijd van 0 tot 30 minuten, en GEEN bandjes met een langere speelduur, deze belasten de 1530 te veel. Nadat we alles aangesloten hebben, kunnen we verder met de computer zelf.

De cursor

Het knipperende vierkantje direct naast het woordje READY wordt de cursor genoemd. De aanwezigheid ervan wil zeggen dat alles wat ingetikt gaat worden daar op het scherm verschijnt. Bij elke toetsdruk schuift de cursor een positie op. Op de voorgaande positie staat het ingetikte karakter weergegeven. Tikt u maar iets in. Kijk hoe hetgeen u intikt op het scherm verschijnt.

Instellen van kleuren

Niets is minder eenvoudig dan het maken van een serie kleuren op het scherm. Daarmee is het instellen van de TV dan mogelijk. Zelfs wanneer u nog totaal niet vertrouwd bent met de computer, hoeft u geen angst te hebben. Volg de aanwijzingen: Aan de linker bovenzijde van het toetsenbord vindt u de toets gemerkt <CTRL>. <CTRL> is de afkorting van ConTRoL. Samen ingedrukt met een andere toets krijgt de computer de opdracht om een bepaalde taak uit te voeren.



Probeer maar eens de <CTRL> toets samen met de toets 9 in te drukken (altijd eerst <CTRL>, dan de andere toets). Gebeurt er iets? Niets, zult u zeggen. Druk nu op een of andere letter. Het teken komt nu in z.g. reverse (omgekeerd) op het scherm in plaats van op de normale manier. Hou nu uw vinger op de spatiebalk. Als u het bovenstaande juist heeft uitgevoerd, verschijnt er een lichtblauwe balk op het scherm die langer wordt en zich voort zal zetten op verdere regels, net zolang als u uw vinger op de balk houdt.

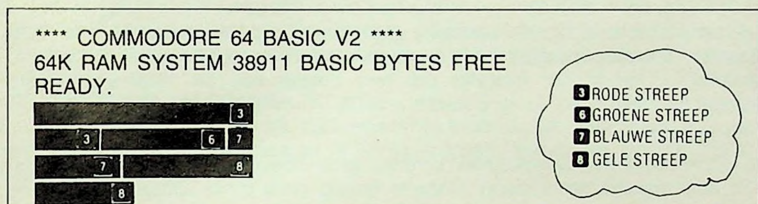
```
**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY.
```

Druk nogmaals op <CTRL> maar nu tegelijkertijd op een van de andere toetsen 1 t/m 8. Vooraan op die toetsen ziet u in het Engels de kleuren staan:

BLK = zwart, RED = rood, PUR = paars,
BLU = blauw, WHT = wit, CYN = cyaan,
GRN = groen, YEL = geel.

Vanaf nu is alles dat op het scherm komt in die kleur. <CTRL> en 8, gevolgd door drukken op de spatiebalk geeft een gele balk! U begrijpt dat op deze wijze elk van de 8 kleuren in balkvorm gemaakt kan worden.

Maak een "testbeeld" en stel dan de TV af door draaien aan de contrastregelaars en de regelaar voor de kleurverzadiging. Onderstaande afbeelding is wat er nu op het scherm zou kunnen staan.



Met behulp van het voorgaande kunt u de instellingen optimaliseren. De nu volgende hoofdstukken zullen u inwijden in de geheimen van de BASIC programmeertaal. Nu uw computer en TV goed zijn ingesteld kunt u echter ook direct gebruik maken van game cartridges, of programma's op tape. Bij elk van de cartridges of programma tapes vindt u een bijsluiter waarop aangegeven staat hoe u te werk dient te gaan. U hoeft eigenlijk niets van programmeren af te weten! Toch raden wij u aan de moeite te nemen de eerste paar hoofdstukken door te lezen. Het verhoogt de kennis van de werking van uw aanwinst.

De Commodore 64 en haar accessoires

De Commodore 64 kent een groot aantal randapparaten die de mogelijkheden van uw computer sterk vergroten. Deze randapparaten omvatten :

- opslag apparatuur
- printers en plotters
- monitoren
- modems voor telecommunicatie
- spel modules
- spraak en grafische modules

OPSLAG APPARATUUR
CASSETTE RECORDER
Het eenvoudigste apparaat waarmee u gegevens opslag kunt realiseren is de 1530 cassette recorder, hiermee kunt u programma's wegschrijven en inlezen van cassette's. Ook is het mogelijk om met behulp van de 1530 eenvoudige sequentiële bestanden op te bouwen.

DISK-DRIVES
De Commodore disk-drives stellen u instaat grote hoeveelheden informatie op een 5.25 inch diskette op te slaan. Diskette opslag biedt u het voordeel van snelle gegevens opslag en opvragen. Alle bestanden en programma's die u op de diskette geschreven heeft worden automatisch in een 'directory' of inhoudsopgave zichtbaar gemaakt. Welke u op scherm of op uw printer af kunt laten drukken.

Op uw Commodore 64 kunt u maximaal 4-1541 disk-drives aansluiten door ze te 'daisy-chainen' dit is het zg. doorlussen.

Ook kunt u gebruikmaken van de disk-drives van de professionele Commodore systemen zoals de SFD-100l en 8250 disk-drives, deze kunt u aansluiten met behulp van een zg. IEEE-interface.



PRINTERS EN PLOTTERS

PRINTERS

Op uw Commodore 64 kunt u de Commodore matrix-printers MPS-801 (een grafische 80 koloms printer) en de MPS-802 (een 80 koloms karakterprinter) aansluiten. Verder kennen we de DPS-1101 margrietwiel printer en de MCP- 801 kleuren printer.

Ook kunt u gebruikmaken van de printers van de professionele Commodore systemen zoals de Commodore 6400 letter kwaliteit printer of de snelle 8023 printer. Deze printers kunt u aansluiten met behulp van de zg. IEEE- interface cartridge.

DE 1701/2 MONITOR

De Commodore 14" (35 cm) geeft u een perfecte kleuren weergave, met een hoog oplossend vermogen, van uw Commodore 64 beeld. De monitor wordt op de Commodore 64 met behulp van een bijgeleverde kabel aangesloten.

MODEMS VOOR TELECOMMUNICATIE

Voor de Commodore 64 zijn VIDITEL interfaces beschikbaar, waarmee u van uw Commodore 64 de goedkoopste VIDITEL-terminal maakt die op de markt verkrijgbaar is.

Met behulp van een 300/300 baud modem kunt u met uw Commodore 64 verbindingen maken naar andere computers of zelfs gebruikmaken van computer data-banken.

DE Z-80 MICROPROCESSOR EN HET CP/M® OPERATING SYSTEEM.

Het plaatsen van de Z-80 microprocessor, maakt uw Commodore 64 tot een dual-microprocessor personalcomputer. Het gebruiken van de Z-80 betekent de toegang tot het populaire CP/M operating systeem waarvoor een grote hoeveelheid software toepassingen, zoals tekstverwerking, veel gebruikte zakelijke en technische programma's, programmeertalen zoals COBOL, FORTRAN etc. en andere nuttige programma's binnen uw bereik komen.

UITBREIDINGEN VOOR SPEL EN ANDER GEBRUIK

Commodore heeft joysticks (stuurknuppels), en paddles (draaiknoppen) die het spelen op uw computer verrijken. Deze uitbreidingen worden niet alleen gebruikt in spelen, maar ook bv. in de programma serie MAGIC-DESK, waarin met behulp van de joystick de computer opdracht wordt gegeven om ingewikkelde handelingen uit te voeren, zoals het afdrukken van een tekst, of het openen van een bureau-lade.

DE COMMODORE SPRAAK MODULE

De Commodore spraak module laat uw Commodore 64 spreken. De module heeft een ingebouwde (Engelse) woordenschat van 235 woorden. Door de module softwarematig te beinvloeden, kan de klank en uitspraak veranderd worden. Het is erg eenvoudig de module vanuit BASIC te programmeren. De module wordt ook door programma's zoals MAGIC-DESK, GORF en WIZARD OF WOR aangestuurd.

PRINTER/PLOTTER
De Commodore 1520 printer/plotter maakt grafieken in 4-kleuren (rood- groen-zwart-blauw). Met de 1520 kunt u staaf diagrammen, cirkel diagrammen en andere complexe x-y grafieken produceren.

COMMODORE GRAFISCHE UITBREIDINGEN
Commodore heeft een variëteit aan grafische programmeer hulpmiddelen, waaronder de Super-Expander cartridge, welke eenvoudige gemakkelijk te leren plot-instructies aan uw Commodore 64 toevoegt, tot de SIMONS BASIC cartridge, welke 118 verschillende BASIC instructies (van grafische instructies tot programmeer hulpmiddelen) kent. En natuurlijk programma's zoals PILOT en LOGO met TURTLE graphics.

HOOFDSTUK 2
HET GROTE BEGIN

HOOFDSTUK 2
HET GROTE BEGIN

- Toetsenbord
- Terug naar normaal schrift
- Laden en saven van programma's
- Print en rekenen
- Rekenvolgorde
- Samenstellen van teksten

Toetsenbord

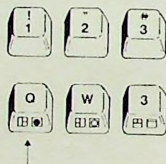
Het meeste verkeer tussen u en de computer zal via het toetsenbord gaan lopen. Het is dus zinvol om even op ons gemak te kijken wat er allemaal mee te doen is. Het toetsenbord ziet er zo ongeveer uit als dat van een schrijfmachine. U komt echter nog een paar extra toetsen tegen waarmee speciale functies kunnen worden opgeroepen. We zullen nu volstaan met het in kort aangeven van de belangrijkste ervan. Uitgebreidere beschrijvingen vindt u verderop in dit boek.

RETURN

Het indrukken van de <RETURN> toets is het teken voor de computer dat de laatst ingetikte informatie gelezen moet worden. Eventueel wordt de informatie in het geheugen opgeslagen.

SHIFT

Ook op een schrijfmachine komen we de <SHIFT> of wagenwissel toets tegen. Veel van de aanwezige toetsen kunnen meerdere tekens laten verschijnen. Zo zal de SHIFT toets samen met een lettertoets ingedrukt, een grafisch teken geven. De lettertoets alleen geeft een (hoofd)letter. De Commodore 64 kan ook grote- en kleine letters weergeven, naast de karakterset van grote letters en grafische tekens. Dat omschakelen gebeurt door indrukken van <SHIFT> en de toets gemerkt C= tegelijkertijd (zie pag. 23).



Los van het normale toetsenbord ziet u nog 4 functietoetsen. Via die 4 toetsen zijn 8 (4 zonder SHIFT, 4 met gelijktijdig indrukken van SHIFT) bereikbaar. Later meer over die functies.

HET VERBETEREN VAN VERGISSINGEN

Vergissen is menselijk en de Commodore 64 houdt daar rekening mee. Op het toetsenbord bevinden zich een aantal toetsen waarmee correcties aangebracht kunnen worden in reeds getikte tekst.

CRSR

Rechts onderaan vindt u 2 toetsen waarop het woordje <CRSR> (CuRSoR) voorkomt. Weet u nog wat dat was? (pag. 10). De ene toets is voorzien van pijlen naar boven en naar beneden en de andere van naar links en rechts wijzende pijlen. U begrijpt al wat daarmee gedaan kan worden. Zonder <SHIFT> gaat de cursor naar beneden of naar rechts, bij

indrukken van de toets met <SHIFT>, naar boven of naar links. Zolang de toets blijft ingedrukt, blijft de cursor doorlopen. We noemen dit verschijnsel met een goed Nederlands woord key-repeat.

INST/DEL

Rechts bovenaan zit een toets gemerkt INST/DEL (INSerT/DELete). Het indrukken van die toets heeft tot resultaat dat het teken juist links van de cursor wordt "opgegeten". Wanneer de cursor midden in een regel of een stuk tekst staat, wordt alles wat rechts van de cursor staat een positie naar links opgeschoven. Omgekeerd <SHIFT> en <INST/DEL> voegt een spatie tussen de cursorpositie en de eventuele volgende tekst. Als u tijdens het intikken bemerkt dat u aan het begin van de regel een teken vergeten bent, dan wandelt u er maar naar toe <SHIFT> <CRSR>, u drukt op <SHIFT> en <INST/DEL> om een spatie tussen te voegen en u tikt het ontbrekende karakter in.

CLR/HOME

<CLR/HOME> rechtsboven ingedrukt verplaatst de cursor naar de z.g. home positie, dat is helemaal links bovenaan in het scherm. <SHIFT> en <CLR/HOME> veegt het hele scherm schoon en zet de cursor in de home positie.

RESTORE

<RESTORE> is de toets welke u nodig heeft om alle door u aangebrachte veranderingen aan scherm, geluid, kleuren, weer ongedaan te maken. Meer over die toets vindt u verderop in het boek.

FUNCTIE TOETSEN

Los van de overige toetsen bevinden zich 4 functie toetsen met in totaal 8 aanroepbare functies (<SHIFT>, weet u wel). U bepaalt zelf welke functies deze toetsen zullen krijgen.

Bijzondere functies

CTRL (ConTRoL)

Het kiezen van de kleuren en nog een aantal andere speciale functies worden verricht door het indrukken van de <CTRL> toets (eerst!) gevolgd door een tweede toets. Eerder al gebruikten we de <CTRL> toets en de toetsen 1 ... 8 voor het kiezen van de kleuren van ons "testbeeld".

RUN/STOP

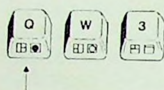
Lopende programma's worden normaliter onderbroken bij indrukken van de RUN/STOP toets. De computer krijgt het bevel STOP! Samen met <SHIFT> geeft de functie <RUN>: het automatisch laden van tape van een programma, gevolgd door het starten ervan.

COMMODORE TOETS



Een van de functies van de C= toets is het omschakelen tussen de beide karaktersets. Bij het inschakelen van de computer worden er op het scherm hoofdletters of grafische tekens weergegeven (al of niet SHIFT ingedrukt). Een druk op een lettertoets geeft op het scherm een hoofdletter. Bij het indrukken van zo'n toets samen met <SHIFT> verschijnt het grafische teken dat rechts op de voorzijde van de toets staat

afgebeeld. Het omschakelen naar de karakterset grote/kleine letters, gebeurt door het indrukken van C= en <SHIFT> tegelijkertijd. Een toets geeft u een kleine of grote <SHIFT> letter, of samen met C= ingedrukt, het grafische teken links voorop de toets.



U komt terug naar grote letters en grafische tekens door hernieuwd indrukken van <SHIFT> en C=.
Een andere functie is het toegankelijk maken van nog eens 8 kleuren. C= samen met een van de toetsen 1 ... 8 geeft de kleur zoals vermeld in het lijstje in hoofdstuk 5.

TERUG NAAR NORMAAL SCHRIFT

Voor het ogenblik hebben we voldoende geoefend met het toetsenbord. Het is nu tijd om te beginnen aan het verkennen van de vele mogelijkheden van de Commodore 64.
De balken die we gemaakt hebben, verwijderen we door het indrukken van <SHIFT> en <CLR/HOME>. Zoals we hierboven zagen komt de cursor terug in HOME positie. De kleur van de letters wordt weer licht blauw als de toetsen XX en 7 worden ingedrukt. Een ding niet vergeten: CTRL en 0 (een streep erdoor om aan te geven dat het om een nul gaat, en niet om de letter O) is nodig om de tekst weer normaal te laten verschijnen. We hadden immers <CTRL> 9 gebruikt om de tekst in reverse te laten verschijnen! Probeert u nog maar eens welk effect het geeft met het afdrukken van spaties na intoetsen van <CTRL> 9 en <CTRL> 0.

TIP

De bovenstaande manier was de moeilijke methode. Het is eenvoudiger om <RUN/STOP> en <RESTORE> tegelijkertijd in te drukken. U zult zien dat de kleuren weer veranderen in licht- en donkerblauw. Het scherm is leeg op het woordje READY na. Programma's worden niet aangetast. Vooral voor hen die veel programmeren is deze methode sneller en handiger. De Commodore 64 kan ook terug ingesteld worden in de oorspronkelijke stand door het intikken van SYS 64759 gevolgd door <RETURN>. Pas op! Alle in de computer aanwezige programmatuur is na gebruik van dit commando uitgewist.

Laden en saven van programma's

Natuurlijk biedt de Commodore 64 de mogelijkheid om programma's vanaf tape of disk te laden of weer weg te schrijven (het z.g. saven spreek uit: seven). U kunt dus programma's vastleggen op tape voor later gebruik. Zo kunt u ook programma's inladen die aangeleverd worden op tape of op disk. Verbind uw 'Datasette' recorder of de diskdrive zoals voorgeschreven (zie pag. 14).

Het laden van programmatuur uit de handel

Kant en klare programmatuur kan aangeleverd worden
1. in z.g. cartridges
2. op cassettetape
3. op diskette.

1. **CARTRIDGES**
Er zijn al heel wat programma cartridges verkrijgbaar, zowel op zakelijk terrein als spellen. De spelletjes :Zjn echte video spelen met kleur en geluid, zoals u ze ook in speelautomaten tegenkomt. Voor het laden van dergelijke programmatuur moet u eerst uw computer uitschakelen en het cartridge achter in de computer steken. PAS OP! ook het verwijderen van cartridges mag niet gebeuren met ingeschakelde computer, tenzij u uw cartridge of computer wilt beschadigen. Schakel TV en computer weer aan en druk op de start toets. (Meestal een van de F-toetsen, raadpleeg de gebruiksaanwijzing van het cartridge).

2. **CASSETTE TAPES**
Sluit de Datasette recorder aan met UIT-geschakelde computer. Schakel de computer weer aan, plaats cassette en wind de tape helemaal terug tot het begin van kant 1. Tik in:

```
LOAD
```

Druk daarna op <RETURN> om aan te geven dat de computer het commando LOAD moet lezen en uit moet voeren. De computer antwoordt met:

```
PRESS PLAY ON TAPE
```

hetgeen betekent dat u de knop gemerkt PLAY op de recorder in moet drukken. Het scherm zal nu leeg worden totdat er een programma is gevonden op de band. De computer zegt dan FOUND gevolgd door de naam van het programma. Druk nu op de C= toets. Het scherm wordt leeg en het laden begint. Men kan onderbreken door het indrukken van RUN/STOP.

3. **DISKETTES**
Haal de diskette uit het envelopje. PAS OP! Raak het oppervlak van de schijf niet aan. Steek de schijf in de disk unit met het etiket bovenop. De ovale uitsparing in het oppervlak verdwijnt het eerst in de gleuf. Aan een van de zijden van de diskette zit een uitsparing, soms met een stukje plakband er overheen. Die uitsparing hoort links van u te zitten. Sluit het deurtje voorzichtig. Tik in:

```
LOAD "*",8 <RETURN>
```

De disk-drive gaat lopen, en op het scherm verschijnt:

```
SEARCHING FOR PROGRAMMA NAAM
LOADING
READY.
```


Na laden zegt de computer READY. De cursor is weer terug, dus de computer wacht op een bevel: tik in

```
RUN <RETURN>
```

Het programma start.

Het laden van cassette van door u geschreven programma's

Het laden van uw eigen programma's is net zo eenvoudig als het laden van programmatuur uit de handel. Wind de tape terug naar het begin. Tik in:

```
LOAD"PROGRAMMA NAAM"
```

en druk op <RETURN> Het is geen ramp als u zich niet meer herinnert hoe de naam ook weer was. U kunt volstaan met LOAD <RETURN>. De computer laadt in dat geval het eerste het beste programma dat op de band staat.

```
PRESS PLAY ON TAPE
```

betekent dat u nu de toets gemerkt PLAY moet indrukken. Na indrukken van die toets wordt het scherm leeg. Het scherm blijft leeg totdat er een programma is gevonden.

```
FOUND PROGRAMMA NAAM
```

de computer wacht nu op een verder teken van u om door te gaan met het eigenlijke laden. Druk op de C= toets. Weer wordt het scherm leeg totdat het laden is voltooid. De computer zegt dan READY en wacht op verdere commando's. Indien het gevonden programma niet het juiste is (op een band kunnen meerdere programma's worden opgenomen) of u wilt toch niet laden, druk dan op <RUN/STOP>.

Het laden van disk van door u geschreven programma's

Het laden van programma's op disk gaat op zowat dezelfde manier:

```
LOAD"PROGRAMMA NAAM",8 <RETURN>
```

De ,8 achter de naam dient om aan te geven dat het laden dient te gebeuren van randapparaat nummer 8, de diskdrive. Na het indrukken van de <RETURN> toets begint de disk te draaien. Het scherm ziet er zo uit:

```
SEARCHING FOR PROGRAMMA NAAM
LOADING
READY.
```

N.B.: Na het inladen van een programma van tape of disk is eventueel nog aanwezige programmatuur in de computer verdwenen en overschreven. Let daar goed op voor u begint! Na laden van een BASIC programma kunt u het laten lopen (RUN-commando) of bekijken (LIST-commando). U kunt elke gewenste verandering aanbrengen en het dan weer SAVEN.

Het wegschrijven van programma's naar tape

Het bewaren van programma's op tape gaat als volgt:

```
SAVE"PROGRAMMA NAAM" <RETURN>
```

Het woordje programmanaam kan elke tekst zijn van maximaal 16 karakters. Na <RETURN> antwoordt de computer:

```
SAVING PROGRAMMA NAAM
OK
READY.
```

Vergeet niet na het intikken van het commando op <RETURN> te drukken om aan te geven dat het SAVE commando uitgevoerd moet worden.

```
PRESS PLAY AND RECORD ON TAPE
```

Dit betekent dat u tegelijkertijd de RECORD en PLAY knop in moet drukken om het wegschrijven te starten. NIET FORCEREN als u de RECORD knop niet ingedrukt krijgt. Controleer of het niet om een beschermde tape gaat (dan is het beschermnokje achterop de tape eruit gebroken). Het scherm wordt weer leeg, en als het schrijven is afgelopen, verschijnt het beeld weer en de tekst READY met cursor. U kunt nu doorgaan met een ander programma of even pauzeren.

Het wegschrijven van programma's naar disk

Het wegschrijven naar een bestaande schijf is nog eenvoudiger. Tik:

```
SAVE"PROGRAMMA NAAM",8
```

De ,8 is de code om aan te geven dat we werken met de diskdrive. De taperecorder heeft ook wel een nummer (,1) maar dit hoeft u er niet bij te zetten (het mag wel). Na het indrukken van <RETURN> loopt de diskdrive en ziet u op het scherm:

```
SAVING"PROGRAMMA NAAM"
OK
READY.
```

Print en rekenen

De taaie kost over LOAD en SAVE die we u voorschotelden, was nodig om u te leren hoe interessante progamma's bewaard kunnen worden op tape of disk. We gaan nu een paar programma's samenstellen waarmee u dat wegschrijven (juist, met het SAVE commando) kunt oefenen.

PRINT "COMMODORE 64"	tik deze regel in en druk op
COMMODORE 64	RETURN
READY.	Dit heeft de computer geschreven

Tikfouten tijdens het intikken van de eerste regel kunt u herstellen door gebruik te maken van <INST/DEL> toets, of van hetgeen u geleerd had op pag. 23. Wat is er nu gebeurd?
Om te beginnen heeft u op regel 1 aangeduid om af te drukken (PRINT) wat u tussen de aanhalingstekens had gezet. Het indrukken van <RETURN> had tot gevolg dat de computer de tekst ging lezen, en het commando uit ging voeren. De computer heeft de tekst COMMODORE 64 afgedrukt op het scherm. Na het uitvoeren van de laatste opdracht zegt de computer bovendien altijd READY.
Het commando PRINT heeft als resultaat dat alles tussen de aanhaaltekens wordt afgedrukt, wat er ook staat. Als de computer antwoordt met:

```
? SYNTAX ERROR
```

kijk dan nog eens goed of u geen tikfout heeft gemaakt of aanhaaltekens hebt vergeten. De computer is heel precies en verwacht van u dat alle instructies op een nauwkeurig omschreven manier worden gegeven. Wanneer u precies de voorbeelden volgt, kan er eigenlijk niets misgaan. Al doende leert men het best. De computer kan echt niet defect raken door tikfouten.
Het PRINT commando is misschien wel het nuttigste en meest gebruikte commando van de BASIC taal. Door middel van dit commando kunt u een scherm vullen met tekst, plaatjes, kleuren of resultaten van berekeningen. Het volgende voorbeeld maakt dat duidelijk. Maak het scherm leeg (SHIFT en CLR/HOME) en tik in:

PRINT 12 + 12	tik deze regel in, en druk op RETURN
24	de computer drukt het antwoord af
READY.	komt omdat er niet meer opdrachten zijn.

De computer kan rekenen! Het resultaat van de optelsom 12 + 12 is door de computer uitgerekend en afgedrukt. Op precies dezelfde manier kunt u aftrekken, vermenigvuldigen, delen... Ook de meer gecompliceerde functies als worteltrekken en machtsverheffen kunnen verricht worden.
Bovendien bent u niet beperkt tot een enkele regel rekenwerk. Het is nu nog even te vroeg om daarop in te gaan. Verderop komen we erop terug.
Het verschil met het bovenste voorbeeld zit hem in de afwezigheid van de

aanhalingstekens, waardoor de computer berekende wat er volgde op het PRINT commando. Wanneer er PRINT "12 + 12" gestaan had, zou de computer ook 12 + 12 hebben afgedrukt in plaats van het resultaat 24. (Probeer maar!)

Rekenkundige bewerkingen

OPTELLEN

Het + teken gaf aan dat we willen optellen. We gaven de intructie: 12 en 12 optellen. De overige rekenkundige bewerkingen worden op gelijkaardige manier aangegeven.
Vergeet niet op <RETURN> te tikken om de bewerkingen uit te laten voeren.

PRINT 45 + 11 + 3	tik deze regel in, en druk op RETURN
59	de computer drukt het antwoord af
READY.	komt omdat er niet meer opdrachten zijn.

AFTREKKEN

Het alom bekende - teken wordt gebruikt voor de bewerking aftrekken:

PRINT 12 - 9	vergeet niet op RETURN te drukken.
3	

VERMENIGVULDIGEN

Het sterretje * duidt de vermenigvuldigingen aan.
12 x 12 wordt dus:

PRINT 12 * 12	vergeet niet op RETURN te drukken.
144	

DELEN

Iets afwijkend van wat u op school leerde. De computer gebruikt de schuine streep /. Het getal 144 gedeeld door 12 wordt dus:

```
PRINT 144 / 12
 12
```

MACHTSVERHEFFEN

Om aan te geven dat het om een macht gaat wordt het teken ↑ gebruikt.
Het getal 12 tot de 5^e macht (12^5) is:

```
PRINT 12↑5
 248832
```

Dit is eigenlijk hetzelfde als het getal 12 vijf keer met zichzelf vermenigvuldigd:

```
PRINT 12 * 12 * 12 * 12 * 12
```

TIP:
In BASIC kunnen veel commando's worden afgekort.
PRINT mag vervangen worden door ? bijvoorbeeld.
U zult steeds meer commando's bijleren. In appendix D staan alle commando's afgekort, en wat er dan op het scherm verschijnt.

Meerdere berekeningen op een regel

Het laatste voorbeeld laat ook zien dat er meer bewerkingen op een regel voor kunnen komen. Bewerkingen kunnen natuurlijk ook gemengd door elkaar voorkomen.

```
? 3 + 5 - 7 + 2          het vraagteken vervangt
 3                       het woordje PRINT
```

Tot nu toe hebben we steeds gewerkt met kleine getallen, en de voorbeelden waren eenvoudig. De Commodore 64 kan daarentegen ook complexe berekeningen met gemak aan. Probeert u eens een aantal grote getallen bij elkaar op te tellen. LET OP! De computer accepteert alleen punten waar wij gewend zijn komma's te schrijven. 12,5 moet in de computer ingegeven worden als 12.5

```
? 123.45 + 345.78 + 7895.687
 8364.917
```

Dat klopt wel ongeveer, niet? Probeer nu het volgende:

```
? 12123123.45 + 345.78 + 7895.687
 12131364.9
```

Als u nu de moeite neemt om het bovenstaande voorbeeld met de hand uit te rekenen zult u merken dat het resultaat van de Commodore 64 afwijkt.
Hoe kan dat nu? De verklaring is dat de computer wel een groot reken-vermogen heeft, maar beperkt is in het aantal cijfers dat in een keer verwerkt kan worden. De Commodore 64 kan getallen tot 10 cijfers manipuleren. Bij het uitprinten echter wordt een maximum van 9 cijfers weergegeven.
Het eindresultaat van ons laatste voorbeeld was afgerond tot 9 cijfers totaal. De Commodore rondt naar boven af als het laatste cijfer 5 of hoger is, en anders naar beneden.
Alle waarden tussen 0.01 en 999 999 999 worden op normale wijze afgedrukt. Buiten dit gebied worden getallen afgedrukt in exponentiele notatie. Exponentiele notatie is het weergeven van zeer grote of zeer kleine getallen als een macht van 10.
Tik in:

```
?123000000000000000 (15 nullen) RETURN
 1.23 E + 17
```

1.23 * 10 ↑ 17 is het resultaat. 1.23 E + 17 maakt het getal begrijpelijker. Zelfs in exponentiele notatie is er een boven- en onderlimiet aan de getalgrootte. Het grootste getal dat de computer kan verwerken is +/- 1.70141183 E+38 en het kleinste is +/- 2.93873588 E-39 (lees +/- als positief of negatief)

Rekenvolgorde

U heeft tijdens het uitproberen misschien niet steeds de eindresultaten gekregen welke u verwachtte. Dat komt omdat de computer berekeningen doet in een bepaalde volgorde. De uitkomst van de berekening:

```
?20 + 8 / 2
```

staat voor u niet vast. Is het 24 of is het 14? U moet hier weten in welke volgorde het rekenen plaatsvindt. 20 optellen bij 8/2 (oftewel 4) geeft 24. Indien u echter eerst 8 bij 20 optelt, en dan het resultaat door 2 deelt, is de einduitkomst 14. Tik het voorbeeld in en kijk.
De reden dat u 24 kreeg is omdat Commodore de rekenregel hanteert die op school wordt onderwezen.

1) – Het minteken voor negatieve getallen eerst
2) ↑ Machtsverheffen, van links naar rechts
3) * / Vermenigvuldigen en delen, van links naar rechts
4) + – optellen en aftrekken, van links naar rechts

Wanneer u het voorbeeld naloopt aan de hand van de rekenregels, constateert u dat eerst de deling wordt uitgevoerd, en vervolgens de optelling. Oefen met wat eigen voorbeelden, en probeert u eerst zelf te voorspellen wat het resultaat zal worden.
Net als op papier zijn ook bij de computer haakjes de manier om een bepaalde rekenvolgorde van te voren op te geven.
U wilt uitrekenen 35 gedeeld door 5 plus 2, en u tikt in

```
? 35/5+2
 9
```

Dan zit u er mooi naast. De computer deelt 35 door 5, en telt 2 bij het resultaat op!
Wat u wilde was het volgende:

```
? 35/(5+2)
 5
```

Nietwaar? De computer heeft eerst de uitdrukking tussen haakjes uitgerekend en vervolgens de deling verricht. Meer uitdrukkingen tussen haakjes op dezelfde regel worden van links naar rechts berekend:

```
?(12+9)*(6+1)
 147
```

Hier in dit voorbeeld wordt 21 vermenigvuldigd met 7, resultaat 147.

Het afdrukken van teksten en getallen.

U vindt misschien wel dat we te lang bezig blijven met zaken die op het eerste gezicht onbelangrijk lijken. Wanneer u gaat beginnen met het echte programmeerwerk, zult u begrijpen waarvoor dit alles dient, en wat voor grote waarde het heeft. We hebben geleerd hoe we getallen op het scherm kunnen zetten, en ook hoe teksten afgedrukt moeten worden. Gesteld dat u een tekst met een getal moet combineren, wat is dan de methode? Het afdrukken van tekst gebeurt door een print commando te laten volgen door tekst tussen aanhalingstekens (pag. 28), getallen door print, gevolgd door het getal of een berekening. Het combineren van tekst en getallen gaat als volgt:

```
PRINT "5 * 9 = "; 5 * 9      de ; geeft aan dat er geen spatie
 5 * 9 = 45                  tussenkomt.
```

Het enige dat we eigenlijk gedaan hebben is het samenvoegen van beide vormen van het PRINT commando. Het eerste stuk zorgde ervoor dat "5 * 9 = " precies werd afgedrukt zoals het er stond en het tweede deel achter de ; deed de berekening en drukte het resultaat af. De ; was de scheiding tussen het tekstgedeelte en de berekening.
De afzonderlijke delen van een combinatie PRINT commando moeten altijd gescheiden worden door een separator: de bovengenoemde ; of een komma. Tikt u maar eens een komma in in plaats van de ; en kijk wat er gebeurt.
Voor de nieuwsgierigen onder u willen we verklappen dat er een ; gezet moet worden als er geen extra spaties tussen de delen moeten komen. De komma echter geeft extra spaties. Probeer:

```
? 2,3,4,5,6                  RETURN
 2    3    4    5
 6
```

De getallen staan verdeeld over de gehele schermbreedte, en op de volgende regel.
Het scherm van de Commodore 64 is opgedeeld in 4 gebieden van 10 karakters. Het gebruik van de komma zet als het ware het volgende deel van het PRINT commando in het volgende "gebied". Het is een soort tabulator functie. Omdat we in het voorbeeld 5 getallen op een regel probeerden te krijgen, ging het 5e getal naar de volgende regel (er zijn immers maar 4 gebieden per regel).
Het verschil in resultaat bij gebruik van ; of , is nuttig voor het samenstellen van complexe schermafbeeldingen, zonder dat het veel moeite kost.

HOOFDSTUK 3
PROGRAMMEREN IN BASIC
HET GROTE BEGIN.

HOOFDSTUK 3
PROGRAMMEREN IN BASIC
HET GROTE BEGIN.

- De volgende stap
- GOTO
- Tips voor het verbeteren van fouten
- Variabelen
- IF...THEN
- FOR...NEXT lussen

De volgende stap

Een enkele regel instructies was tot nu toe het maximum dat we haden ingegeven. Direct na het indrukken van de RETURN toets werd de ingetikte instructie uitgevoerd. Dit is de (directe uitvoering) van commando's. Men komt in de literatuur daarvoor ook wel de Engelse term Immediate mode of Calculator mode tegen. Als wij een wat zwaardere opdracht willen volbrengen, dan moeten we de computer toch wel meer dan een regeltje aan instructie kunnen geven! Zo'n verzameling opdrachten of instructies noemen we een programma. Een programma geeft u pas de mogelijkheid de volle kracht van de Commodore 64 te gebruiken.

Het schrijven van een eerste programma is doodsimpel:
1. maak het scherm leeg (SHIFT en CLR/HOME)
2. type NEW gevolgd door RETURN. Dit is nodig om getallen die overgebleven zijn van het oefenen te kunnen verwijderen.
3. Tik nauwkeurig het onderstaande over. Vergeet niet op RETURN te drukken na het laatste karakter.

```
10 ? "Commodore 64"
20 GOTO 10
```

Geef nu in: RUN en druk op RETURN. Kijk goed naar het scherm: Er wordt constant de tekst "Commodore 64" op geschreven! Als u er genoeg van heeft kunt u op RUN/STOP drukken om het programma weer te stoppen.

Regelnummers

In het programma komen direct een paar belangrijke principes aan de orde. Ten eerste ziet u dat we een getal voor elk commando hebben geplaatst. Dit z.g. regelnummer vertelt aan de computer in welke volgorde de instructies uitgevoerd moeten worden. De regelnummers zijn ook de punten waar een programma naar

toe kan springen. Regelnummers kunnen willekeurig gekozen worden in het gebied 0.....63999.

```
10 PRINT "COMMODORE 64"
↑  ↑
regelnummer instructie
```

```
RUN
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
BREAK IN 10
READY.
```

De computer voert het programma normaliter uit in de volgorde van de regelnummers. Het is een goede gewoonte om de regels te nummeren in veelvouden van 10. Handig als er later instructies tussengevoegd moeten worden. Naast PRINT hebben we ook de instructie GOTO gebruikt. Dit commando stuurt de computer rechstreeks naar het achter GOTO genoemde regelnummer. Het programma gaat verder vanaf dat punt.

```
┌→10 PRINT "COMMODORE 64"
└─20 GOTO 10
```

Ons voorbeeld drukt de tekst "COMMODORE 64" af en gaat vervolgens door naar regel 20. Daar staat echter dat de computer terug moet gaan naar regel 10. Het afdrukken van de tekst gebeurt een tweede keer en .... u begrijpt het al, er komt geen einde meer aan. Gelukkig dat we kunnen ingrijpen met de RUN/STOP toets.

Het list commando

De instructie LIST <RETURN> heeft als resultaat dat het door u ingetikte programma op het scherm verschijnt. Ziet u dat de computer het ? heeft veranderd in het woordje PRINT. Er kunnen desgewenst veranderingen in het programma worden aangebracht. Ook kunt u het programma wegschrijven, (SAVE) of nog eens laten lopen (RUN).
Een van de belangrijkste verschillen tussen een programma en het geven van directe instructies is dat programma's na uitvoeren ervan niet verloren raken. Steeds als u LIST geeft, ziet u uw programma terug.

Tips voor het maken van veranderingen

Er zijn een aantal methoden om fouten tijdens het ingeven van programmaregels te verbeteren of om veranderingen aan te brengen.

1. De hele regel kan opnieuw ingegeven worden. De computer vervangt de foute regel door de (goede) nieuwe.
2. Ongewenste regels worden verwijderd door het regelnummer in te tikken direct gevolgd door RETURN.
3. Het is mogelijk verbeteringen aan te brengen met gebruikmaking van de cursortoetsen en de INST/DEL toets.

Als voorbeeld maken we een tikfout in regel 10 van ons voorbeeld.
Type LIST <RETURN>. Ga met gebruikmaking van de toetsen SHIFT en CURSOR naar regel 10. Vervolgens loopt u met de CURSOR toets naar het punt waar de fout zich bevindt. Tik hier de juiste letter in, gevolgd <RETURN>. De regel is nu vervangen door de gecorrigeerde regel.
Indien er letters tussengevoegd dienen te worden, dan kunt u dit doen door op de gewenste plaats de toetsen SHIFT en INST/DEL te gebruiken. Overbodige karakters verwijdert men door DEL.
Wanneer u na het aanbrengen van de correcties wederom LIST <RETURN> intikt, zult u constateren dat de correcties zijn aangebracht. Bovendien hoeven correcties niet in volgorde te worden uitgevoerd. De computer verzorgt dat voor u.
Verander nu het programma door achter regel 10 een punt komma te plaatsen. RUN daarna weer het programma.

```
10 PRINT "COMMODORE";
20 GOTO 10

COMMODORE   COMMODORE   COMMODORE   COMMODORE
COMMODORE   COMMODORE   COMMODORE   COMMODORE
COMMODORE   COMMODORE   COMMODORE   COMMODORE
COMMODORE   COMMODORE   COMMODORE   COMMODORE
BREAK IN 10
READY.
```

Variabelen

De variabelen vormen een van de meest essentiele en meest gebruikte onderdelen van elke programmeertaal. De variabelen bevatten de informatie in de computer. We zullen trachten duidelijk te maken hoe het een en ander werkt. Die kennis stelt ons in staat gecompliceerde resultaten te verkrijgen, welke onmogelijk op andere wijze verkregen zouden kunnen worden.Stelt u zich in het inwendige van de computer een aantal dozen voor. In elk van die dozen kan een getal zitten, of een stukje tekst. Iedere doos is voorzien van een etiketje met een door ons zelf uit te kiezen naam. Die naam noemen we de variabele. De variabele stelt de inhoud van de doos voor.

Als we bijvoorbeeld zeggen:

```
10 X%   = 15
20 X    = 23.5
30 X$   = "DE SOM VAN X% + X ="
```

dan kent de computer de variabelen

X%	15
X	23.5
X$	DE SOM VAN X% + X

De naam van de variabele stelt de doos, of de plaats in het computergeheugen voor waar de momentele waarde van de variabele ligt opgeslagen. Er zijn drie soorten variabelen:

1) integere variabelen, gekenmerkt door het teken % na de naam. Een integer getal is een heel getal met waarde tussen -32767 en + 32767.
2) Z.g. floating point variabelen, of variabelen met vlottende komma.
3) Z.g. tekst variabelen, of strings. Deze variabelen bevatten geen getal, maar een tekst of een aaneenschakeling van karakters. Dit type variabele is te herkennen aan de $ na de variabele naam.

Niet alle variabele namen zijn geldig.

- Een variabele naam kan uit een of twee tekens bestaan. Het eerste teken is altijd een letter, het (facultatieve) tweede teken is of een letter of een van cijfers 0...9.

- U mag variabele namen gebruiken die uit meer dan 2 karakters bestaan, maar alleen de eerste twee karakters worden door de computer herkend. PAPA en PA zijn voor de computer dus equivalent.

- Geldige integer namen zijn dus b.v. A%, X%, A1% of NM%; het floating point equivalent zou zijn A, X, A1 en NM. Geldige string namen zijn b.v. A$, (Spreek uit A-String) of NM$.

- Variabele namen mogen geen BASIC commando's bevatten, zoals b.v. GOTO, RUN. Achterin het boek (Appendix C) vindt u een complete lijst van alle BASIC woorden.

Om een idee te krijgen hoe de variabelen gebruikt worden, hebben we het volgende voorbeeld:

```
NEW
10 X% = 15
20 X = 23.5
30 X$ = "De som van X% en X ="
40 PRINT "X% ="; X%, "X ="; X
50 PRINT X$; X% + X
```

RUN <RETURN> geeft het volgende resultaat

```
X% = 15        X = 23.5
DE SOM VAN X% + X = 38.5
```

Alle tot nu toe bekende regels hebben we in dit voorbeeld toegepast. U ziet hoe we een beeld kunnen opbouwen rond de som van twee variabelen. In de regels 10 en

20 werd er een integere waarde toegekend aan variabele X% en een floating point waarde aan X. Het getal behorende bij de variabele werd hier als het ware in de doos gestopt. In regel 30 werd een stuk tekst toegekend aan variabele X$. In regel 40 worden de 2 typen PRINT commando's gebruikt om een boodschap af te drukken gevolgd door de actuele waarden van X% en X. In regel 50 tenslotte wordt X$ afgedrukt en de som van X% en X.
De 2 suffixen % en $ zorgen ervoor dat de variabelen hoewel de naam X hetzelfde is, toch uniek worden. (Suffix = achtervoegsel).
X, X% en X$ zijn 3 verschillende variabelen.
Variabelen hebben echter nog veel meer mogelijkheden. Door het veranderen van de waarden, zal de nieuwe waarde de oude waarde in de doos vervangen:

X = X + 1

In de normale algebra zou het bovenstaande onzin zijn. Bij het programmeren is het een van de meest gebruikte formules. Het betekent dat X gelijk wordt aan de oude waarde die X had, vermeerderd met 1.

If ..... then

Gewapend met onze kennis over het veranderen en aanpassen van variabelen kunnen we het volgende programma uitproberen:

```
NEW
10 CT = 0
20 ? "COMMODORE 64"
30 CT = CT + 1
40 IF CT < 5 THEN 20
50 END
RUN
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
READY.
```

Het woordje NEW bent u al op pag. 40 tegengkomen. Door dit commando wordt elk in de computer aanwezig BASIC programma gewist.
In dit voorbeeld komen er 2 nieuwe BASIC commando's om de hoek kijken, die een beetje controle geven over het eindeloos doorlopende programmaatje waarmee we dit hoofdstuk begonnen.
IF .... THEN voegt wat logica aan het programma toe. Het wil zoveel zeggen dat ALS (IF) aan een bepaalde voorwaarde wordt voldaan DAN (THEN) moet er iets worden gedaan. Als NIET aan de voorwaarde wordt voldaan, dan moet direct de volgende BASIC regel worden uitgevoerd. De volgende voorwaarden kunnen worden gebruikt:

Symbool	Betekenis
<	Kleiner dan
>	Groter dan
=	Gelijk aan
<>	Niet gelijk aan
>=	Groter dan of gelijk aan
<=	Kleiner dan of gelijk aan.

Het gebruik is eenvoudig. De kracht van deze condities is verrassend groot:

```
   10 CT = 0
┌→ 20 ? "COMMODORE 64"
│  30 CT = CT + 1
└─ 40 IF CT < 5 THEN 20
   50 END
```

In ons voorbeeld hebben we een lus (LOOP) ingebouwd welke wordt doorlopen onder bepaalde voorwaarden.
In regel 10 wordt de teller op nul gezet. In regel 20 wordt onze boodschap: "COMMODORE 64" afgedrukt. Op regel 30 wordt de variabele CT opgehoogd met 1. Deze regel telt hoe vaak we de lus doorlopen hebben. Elke keer dat we de lus doorlopen geeft het ophogen van de variabele CT met 1. Regel 40 is de stuurregel. Als CT kleiner is dan 5, wat betekent dat we de lus minder dan 5 x doorlopen hebben, dan gaat het programma door naar regel 50, het einde van het programma. Probeer het programma uit. Door het veranderen van de grenswaarde in regel 40 kunt u een willekeurig aantal regels laten afdrukken. Er zijn nog veel meer gebruiksmogelijkheden voor IF ... THEN. Verderop in het boek komt u ze tegen.

For ...... next lussen

Er bestaat een simpeler en populairdere methode om tot hetzelfde eindresultaat te komen. Dit bereiken we met de FOR ... NEXT lus. Bekijk het volgende voorbeeld.

```
NEW
10 FOR CT = 1 TO 5
20 PRINT "COMMODORE 64"
30 NEXT CT

RUN
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
COMMODORE 64
READY.
```

Zoals u kunt zien is het programma een stuk kleiner, en veel directer. CT begint bij 1 (regel 10). Op regel 20 wordt er iets geprint. Op regel 30 wordt CT met een

opgehoogd. NEXT stuurt het programma automatisch terug naar regel 10, naar het FOR gedeelte van het FOR ... NEXT commando. Dit proces gaat door totdat CT de ingegeven bovenste grenswaarde bereikt. De gebruikte variabele kan eventueel in kleinere stapjes worden veranderd. Probeer dit maar eens:

```
NEW
10 FOR NB = 1 TO 10 STEP .5
20 PRINT NB,
30 NEXT NB

RUN
1  1.5     2  2.5
3  3.5     4  4.5
5  5.5     6  6.5
7  7.5     8  8.5
9  9.5     10
```

Op het scherm ziet u de getallen 1 tot 10 in stapjes van 0.5. Het enige dat we hier doen is het afdrukken van de successievelijke waarden van variabele NB. STEP specificeert de stapgrootte. Het is zelfs mogelijk een negatieve stapgrootte aan te geven:

```
10 FOR NB = 10 TO 1 STEP -0.5
```

Hier gebeurt op het scherm net het tegenovergestelde. NB loopt van 10 tot 1, steeds 0.5 minder.

HOOFDSTUK 4
BASIC VOOR GEVORDERDEN

HOOFDSTUK 4
BASIC VOOR GEVORDERDEN

- Inleiding
- Bewegende beelden
- Input
- Get
- De funktietoetsen
- Random getallen en andere functies
- Raadsel
- Uw beurt om te werpen
- Random graphics
  de CHR$ en ASC functie

Inleiding

De volgende hoofdstukken zijn hoofdzakelijk bedoeld voor hen die al min of meer vertrouwd zijn met BASIC, en de basisprincipes kennen welke nodig zijn om meer ingewikkelde programma's te kunnen schrijven. Indien u net begint met leren programmeren kan het zijn dat u zo hier en daar struikelt over de moeilijke woorden. Maar verlies de moed niet! Speciaal voor u hebben we 2 hoofdstukken, vol plezier, geschreven over sprites en geluid, vol eenvoudige voorbeelden voor de nieuwe generatie programmeurs. Die voorbeelden zullen u een goede indruk geven over het hoe en wat van de geluids- en grafische mogelijkheden van uw Commodore 64.
Mocht u besloten hebben een expert te worden in het programmeren in BASIC, dan vindt u in appendix N een literatuuropgave.
Zij die al vertrouwd zijn met BASIC zullen in de volgende hoofdstukken diepergaande voorbeelden aantreffen. Het "Commodore 64 Programmers Reference Manual" bevat volledige informatie over alle aspecten van de Commodore 64. Dit boek is te betrekken via uw dealer.

Bewegende beelden

We zullen nu gaan oefenen met de grafische mogelijkheden van de Commodore 64. Daartoe maken we gebruik van hetgeen we tot nu toe hebben geleerd, aangevuld met wat nieuwe dingen. Type onderstaand programma in en laat het RUNnen. U zult vast opmerken dat wij in een PRINT opdracht ook cursor bewegingen en schermcommando's kunnen meeprogrammeren. Wanneer we in de tekst CRSR LINKS zeggen, bedoelen we dat op die plaats de SHIFT en CRSR toetsen worden ingedrukt. Op het scherm ziet u dan het codeteken voor die toetsdruk verschijnen. (2 verticale streepjes in reverse). Op dezelfde manier zal SHIFT en CLR/HOME op het beeld verschijnen als een hartje in reverse. Om vergissingen in het begin te voorkomen zijn de cursor besturingstoetsen aangeduid met [CLR], [LINKS], [RECHTS], [HOOG] en [LAAG]. Later zullen wij, in de programmalistings, de tekens laten zien zoals u ze op het beeldscherm ziet.

```
10 REM STUITERENDE BAL
20 PRINT"[CLR]"
21 REM "[CLR] IS SHIFT EN HOME"
22 REM "[LAAG] IS CRSR DOWN"
23 REM "[LINKS] IS CRSR LEFT"
25 FOR X = 1 TO 10 : PRINT "[LAAG]" : NEXT
30 FOR BL = 1 TO 40
40 PRINT" Q[LINKS]";:REM "Q IS SHIFT EN Q"
50 FOR TM = 1 TO 5
60 NEXT TM
70 NEXT BL
75 REM BAL VAN RECHTS NAAR LINKS
80 FOR BL = 40 TO 1 STEP -1
90 PRINT" [LINKS][LINKS]Q[LINKS]";
100 FOR TM = 1 TO 5
110 NEXT TM
120 NEXT BL
130 GOTO 20
```


TIP:
Alle woorden in bovenstaande tekst staan in hun geheel op dezelfde regel. Zolang u nog niet op de RETURN toets gedrukt heeft, echter, gaat de cursor na karakter 40 naar de volgende regel. Dit kan midden in een woord gebeuren! De Commodore 64 leest echter het hele woord.

Na RUN ziet u een bal tussen de linker- en rechter schermrand heen en weer stuiteren. Wanneer we het programma goed lezen, kunnen we zien hoe we dit voor elkaar hebben gekregen. REM in regel 10 is alleen maar een opmerking (REMARK) die ons vertelt wat het programma doet. Een REM heeft geen invloed op het programma, maar dient alleen ter verduidelijking.

```
10 REM STUITERENDE BAL
20 PRINT"[CLR]"
21 REM "[CLR] IS SHIFT EN HOME"
22 REM "[LAAG] IS CRSR DOWN"
23 REM "[LINKS] IS CRSR LEFT"
25 FOR X = 1 TO 10 : PRINT "[LAAG]" : NEXT
30 FOR BL = 1 TO 40
40 PRINT" Q[LINKS]";:REM "Q IS SHIFT EN Q"
50 FOR TM = 1 TO 5
60 NEXT TM
70 NEXT BL
75 REM BAL VAN RECHTS NAAR LINKS
80 FOR BL = 40 TO 1 STEP -1
90 PRINT" [LINKS][LINKS]Q[LINKS]";
100 FOR TM = 1 TO 5
110 NEXT TM
120 NEXT BL
130 GOTO 20
```

In regel 20 wordt het scherm schoongemaakt en wordt de cursor op HOME (links bovenin het scherm) gezet. Regel 25 zet de cursor 10 regels naar onderen. De bal komt zodoende in het midden van het scherm te staan. Het weglaten van regel 25 zou tot gevolg hebben dat de bal bovenin het beeld zou bewegen. Regel 30 zet een loop op (naar 70) om de bal over de 40 posities van links naar rechts te bewegen.
In regel 40 wordt veel gedaan. Eerst wordt er een spatie afgedrukt (=niets) over de voorgaande balpositie, vervolgens de bal en tenslotte een stapje met de cursor naar links om alles al klaar te zetten voor het uitwissen van de nieuwe bal.
De loop in de regels 50 en 60 dient om de afloop van het programma (en de bal!) te vertragen. Zonder deze vertraging (we noemen zoiets een wachtlus) zou de bal te snel over het scherm bewegen. Regel 70 maakt de lus af die we in regel 30 waren begonnen. Bij elke keer dat de lus wordt doorlopen, schuift de bal een positie naar rechts op. Binnen de lus 30-70 vinden we een tweede lus, de wachtlus 50-60: dit mogen we doen. We moeten er alleen goed op letten dat de lussen elkaar niet kruisen. het is een goede gewoonte om lussen in te tekenen zoals wij dat deden in het voorbeeld. Draai de commando's in de regels 60 en 70 maar eens om. De computer raakt in de war!
De regels 80 t/m 120 draaien het proces 30-70 precies om. In die regels gaat de bal van rechts naar links. Regel 90 wijkt af van regel 40 omdat we eerst de bal (rechts) moeten uitwissen, en vervolgens naar links moeten lopen.

Als alles eenmaal doorlopen is, gaat het programma terug naar regel 20 en begint alles weer van voren af aan. Keurig! U kunt het programma onderbreken door tegelijkertijd op <RESTORE> en <RUN/STOP> te drukken. Een variatie op het programma is het volgende:

40 PRINT "Q"; om de bal te maken moet u op SHIFT en Q drukken.

RUN en kijk. Omdat we de cursorbeweging weglieten blijft elke bal op het scherm staan totdat hij weggeveegd wordt door de bal in het 2e gedeelte van het programma.

Input

Tot nu toe waren alle variabelen voor starten van een programma ingesteld. Als het programma eenmaal liep, kon er niets meer aan worden veranderd. Met input kunnen we nieuwe informatie aan het programma toevoegen, tijdens het lopen ervan, waarna met die nieuwe informatie verder gewerkt kan worden.
Het werken met INPUT wordt duidelijk aan de hand van het volgende voorbeeld. Type eerst NEW <RETURN> om een vorig programma uit te wissen:

```
10 INPUT A$
20 PRINT "U TIKTE IN: ";A$"
30 PRINT
40 GOTO 10
RUN
? COMMODORE 64
U TIKTE IN: COMMODORE 64
```


De werking is simpel. Er verschijnt een vraagteken met een knipperende cursor. Dit geeft aan dat de computer wacht tot u iets heeft ingetikt. Tik nu iets in, en druk op RETURN. De computer antwoordt dan met: U zei: gevolgd door hetgeen u intikte. Het lijkt allemaal eenvoudig, maar bedenk eens wat u de computer niet kan laten doen met uw informatie! Via INPUT kunt u getal- alsmede stringvariabelen invoeren. Zelfs kunt u via INPUT tekst op het scherm zetten:

INPUT "TEKST"; variabele (TEKST mag maximaal 38 karakters lang zijn)

is de complete, juiste vorm van een INPUT. Ook kan:

Input variabele

In het geval u geen tekst voor het vraagteken wilt zetten.
Ook dit programma kan onderbroken worden door het indrukken van RUN/STOP en RESTORE tegelijkertijd.
Het volgende programma is niet alleen nuttig, maar geeft ook een beeld van wat we tot nu toe geleerd hebben, inclusief het nieuwe INPUT commando:

```
1 REM TEMPERATUUR CONVERSIE
5 PRINT "[CLR]" : REM "[CLR]" IS CLR/HOME
10 PRINT "VAN FAHRENHEIT OF VAN CELSIUS (F/C)";: INPUT A$
20 IFA$ = "" THEN 10
30 IFA$ = "F" THEN 100
40 IFA$ = "C" THEN 50
50 INPUT"GEEF AANTAL GRADEN CELSIUS:";C
60 F = (C*9)/5 + 32
70 PRINT C;" GR. CELSIUS =";F;" GR. FAHRENHEIT"
80 PRINT
90 GOTO 10
100 INPUT"AANTAL GRADEN FAHRENHEIT:";F
110 C = (F-32)*5/9
```

```
120 PRINT F;"GR. FAHRENHEIT =";C;" GR. CELSIUS"
130 PRINT
140 GOTO 10
```

Als u dit programma laat runnen, ziet u - INPUT - aktie! In regel 10 wordt het INPUT commando niet alleen gebruikt om informatie naar binnen te halen, maar ook om de tekst van de vraag af te drukken. U ziet (regel 10, regel 50) ook dat we zowel string- als getalsvariabelen kunnen binnenhalen! In de regel 20, 30 en 40 wordt er gecontroleerd wat we hebben ingevoerd. Als we niets hebben ingevoerd (we hebben alleen op <RETURN> gedrukt), zorgt regel 20 ervoor dat het programma weer opnieuw begint op regel 10. Als er F ingetikt wordt (gevolgd door <RETURN>) weten we dat de gebruiker graden Fahrenheit naar Celsius wil converteren, dus het programma springt naar regel 100 waar die conversie wordt uitgevoerd. Regel 40 doet nog een extra controle. Er zijn immers maar 2 geldige antwoorden: F en C. Als het antwoord gelijk is aan C, dan beginnen we opnieuw in regel 10. Zo'n controle lijkt flauw en overbodig, maar het is toch een goede gewoonte bij het programmeren. Gebruikers die een programma niet kennen, raken in de war wanneer er iets vreemds gebeurt, omdat fouten tijdens het intikken van informatie niet worden herkend.
Wanneer we aan het programma hebben opgegeven welke conversies er uitgevoerd moeten worden, vraagt de computer om het aantal graden en berekent het equivalent. Tenslotte wordt de ingegeven waarde afgedrukt, gevolgd door het equivalent in de andere gradensoort. Het rekengedeelte is recht- toe recht-aan algebra. Na afloop van het rekenen en afdrukken springt het programma terug naar regel 10 voor de volgende gang. Het scherm ziet er als volgt uit bv:

```
VAN FAHRENHEIT OF VAN CELSIUS (F/C) ?F
GEEF AANTAL GRADEN FAHRENHEIT: ?32
32 GR. FAHRENHEIT = 0 GR. CELSIUS

VAN FAHRENHEIT OF VAN CELSIUS (F/C) ?
```

Get

Met GET kunt u een enkel karakter binnenhalen, zonder op de <RETURN> toets te hoeven drukken. Het is een manier om de snelheid van invoer te vergroten. De variabele achter GET neemt de bijbehorende waarde aan van om het even welke toets die ingedrukt wordt.

```
10 GET A$ : IF A$ = "" THEN 10
20 PRINT A$;
30 GOTO 10
```

HIER GEEN SPATIE

Bij runnen van het programma wordt het scherm leeg. Via regel 20 wordt elke toetsdruk op het scherm afgedrukt en wordt vervolgens een nieuw karakter binnen ge-GET. Karakters welke via GET binnenkomen, worden normaal niet afgedrukt, tenzij we zoals hier een PRINT opdracht geven!
De structuur van regel 10 is ook zeer belangrijk. GET werkt constant, zelfs als er geen toets wordt ingedrukt. (Input wacht op een antwoord!) Het tweede deel van

regel 10 zorgt ervoor dat we op regel 190 blijven als er niets ("", een z.g. lege string, niets) ingedrukt was. Verwijder dit gedeelte maar eens en zie wat er gebeurt. RESTORE en RUN/STOP samen stoppen het programma.
Het eerste gedeelte van het conversieprogramma van pagina 43 kan als volgt herschreven worden voor gebruik met GET:

```
10 PRINT "VAN FAHRENHEIT OF VAN CELSIUS (F/C)"
20 GET A$ :IF A$ = "" THEN 20
30 IF A$ <> "C" THEN 20
```

Een soepeler programmaverloop is het gevolg:
Er gebeurt niets tenzij de gebruiker een juist antwoord (F of C) ingeeft. U kunt natuurlijk deze nieuwe versie ook weer SAVEN en bewaren voor later gebruik!

Functie toetsen

Zoals u zich kunt herinneren uit het vorige hoofdstuk, hebben we verteld dat de toetsen aan de rechter zijde van de computer (F1 t/m F8) 'geprogrammeerd' kunnen worden om een speciale functie te verrichten.

In het navolgende stukje zullen we laten zien hoe dit eenvoudig te realiseren is.

1. We maken gebruik van de GET instruktie om het toetsenbord uit te lezen.

2. We maken gebruik van het IF statement om te vergelijken of de ingedrukte toets overeenkomt met de CHR$ kode van de functietoets die we willen gebruiken. Ieder teken op het toetsenbord heeft een eigen unieke CHR$ waarde (0-255). De CHR$ kode voor F1 is bijvoorbeeld 133. In appendix F is een lijst opgenomen van alle toetsen en hun overeenkomstige CHR$ waarden.

3. Gebruik het THEN statement om de computer te vertellen wat de functie toets BINNEN het BASIC programma moet doen.

Wanneer het BASIC programma geRUN'd wordt, zal dus het gedeelte achter THEN bepalen wat de functie van de toets werkelijk geworden is.

Bijvoorbeeld :

```
10 GETA$:IFA$=""THEN 10
20 IFA$=CHR$(137)THEN A$=CHR$(14)
30 IFA$=CHR$(134)THEN A$="DE HEER"+CHR$(13)
40 IFA$=CHR$(140)THEN STOP
50 IFA$=CHR$(139)THEN 2500
60 PRINTA$
70 GOTO 10
2500 REM VERVOLG PROGRAMMA
2510 ...
```

In regel 10 kijken we of er een toets ingedrukt wordt. Als er geen toets wordt ingedrukt is A$ 'leeg' en kijken we opniew met de GET of er een toets is ingedrukt. Pas als er een toets is ingedrukt gaan we naar regel 20.
Regel 20 verteld het programma dat de CHR$ waarde 14 aan A$ toegekend moet worden als F2 ingedrukt wordt. CHR$ 14 is de 'schakelaar' die het beeldscherm van kleine/hoofdletters omschakeld naar hoofdletters/grafische tekens.
In regel 30 wordt functie toets 3, CHR$(134), gebruikt om A$ te vullen met de tekst "DE HEER" en CHR$(13). CHR$(13) is de kode voor de RETURN toets. Wordt F8 ingedrukt dan zal het programma stoppen, en als F7 ingedrukt wordt, zal het programma verder gaan met het uitvoeren van programma regel 2500.

De CHR$ code's voor de 8 functietoetsen zijn :

F1 = CHR$(133) F2 = CHR$(137)
F3 = CHR$(134) F4 = CHR$(138)
F5 = CHR$(135) F6 = CHR$(139)
F7 = CHR$(136) F8 = CHR$(140)

De Commodore 64 Programmers Reference Guide geeft u meer informatie over hoe U de functie toetsen kunt programmeren. Deze handleiding kunt u bij uw Commodore dealer verkrijgen.

In somige programma's wordt veelvuldig gebruikgemaakt van de functie toetsen. De uiteindelijke werking van deze toetsen is vaak verschillend en geheel bepaald door het programma dat op dat moment in de computer actief is.

Met de BASIC uitbreidings cartridge SIMONS BASIC (C64108), is het mogelijk om de functie toetsen zodanig te programmeren dat de toetsen ook in direct-mode te gebruiken zijn. Zodat aan een bepaalde toets een BASIC statement gekoppeld kan worden. (bv. LIST, RUN, PRINT etc.).

Random getallen en andere functies

De Commodore 64 bevat een aantal functies voor het verrichten van bijzondere taken. U kunt zich zo'n functie voorstellen als een soort ingebouwd BASIC programma. In plaats van het intypen van (soms vele) commando's voor het verrichten van een speciale berekening kunt u volstaan met het aanroepen van de functie door middel van een speciaal commando. De computer doet de rest voor u.
Het zal vaak gebeuren, vooral bij het ontwerpen van spellen of educatieve programma's, dat u een of ander willekeurig getal nodig hebt. Dit kan zijn voor simulatie van het werpen van een dobbelsteen bijvoorbeeld. Natuurlijk kunt u een programma schrijven dat dergelijke getallen genereert maar het is eenvoudiger om de RANDOM (RND) functie aan te roepen voor het maken van zo'n willekeurig (RANDOM) getal. Het volgende programma illustreert dit:

```
10 FOR X = 1 TO 10
20 PRINT RND(1),
30 NEXT
```

ALS U DE KOMMA WEGLAAT WORDT ALLES ONDER ELKAAR AFGEDRUKT

Na RUN ziet u:

```
.712457814      .214578124
.478941234      3.67942478E-04
.147325542      .716714662
.447822234      .614877177
.237145542      .198745662
```

Wat? Heeft u andere getallen dan wij? Goed zo! Als dat namelijk niet het geval was, zou het er niet best uitzien. De getallen moeten namelijk volkomen willekeurig zijn. Laat het programma een paar maal runnen. U zult zien dat u steeds andere resultaten krijgt. Hoewel de getallen zelf willekeurig zijn, zullen u toch een paar dingen opvallen:

1) De getallen zijn altijd groter dan 0, maar nooit groter dan 1 dus 0 < getal < 1.
2) Het zijn derhalve altijd gebroken getallen.

Omdat het RND-getal altijd kleiner is dan 1 hebben we er in deze vorm niet veel aan om het werpen van een dobbelsteen te simuleren. Bovendien willen we vaak hele getallen simuleren en geen breuken.
Er zijn een paar eenvoudige manieren om RND getallen te laten genereren in het gewenste getalsgebied.
Vervang regel 20 door het onderstaande en RUN opnieuw.

```
20 PRINT 6*RND(1),

RUN
3.15457814      2.42878124
5.14871234      4.67942481
1.29725542      5.24781662
2.35478542      4.26874562
0.25475542      1.24545442
```

Dit mag dan wel het probleem hebben opgelost van de begrenzing 0 < getal < 1, maar we zitten nog steeds met de afronding. We roepen nu nog een functie aan, de INTeger functie. Deze functie (INT) converteert gebroken getallen naar integere getallen. (De integere waarde van een getal is het hele getal kleiner dan of gelijk aan het getal;
dus INT (1.7) = 1, INT (-1.7) =-2, INT (100) = 100.
Vervang regel 20 door het volgende en RUN weer:

```
20 PRINT INT(6*RND(1))

RUN

2   2   3   4
1   5   2   0
1   4
```

Dat loste al veel op! We zijn al een stuk dichter bij de oplossing van ons probleem. Het genereren van een willekeurig getal tussen 1 en 6. U ziet echter dat de resultaten liggen tussen 0 en 5. Tel tenslotte 1 op bij de calculatie in regel 20, dus:

```
20 PRINT INT(6*RND (1) + 1
```

Dit is het resultaat dat we wilden bereiken.
Binnen de haakjes van de INT functie (het z.g. antwoord) kan een getal, een variabele of weer een functie voorkomen. De bovengrens van getalgeneratie d.m.v. RND wordt ingesteld door het RND-getal te vermenigvuldigen met die bovengrens. Als u bijvoorbeeld getallen tussen 1 en 25 wilt genereren, doet u:

```
20 PRINT INT (25*RND (1) +1)
```

De algemene formule voor het genereren van willekeurige getallen binnen bepaalde getalswaarden is

```
GETAL = INT (ONDERGRENS+(BOVENGRENS-ONDERGRENS+1)*RND(1))
```

Een raadseltje

Zullen we proberen onze kennis van RANDOM getallen in praktijk te brengen? Het volgende programma geeft niet alleen een goed beeld van het gebruik van RND, maar laat u ook kennis maken met wat theorie rond programmeren. Tijdens het lopen van het programma wordt een RANDOM getal NM gegenereerd.

```
1 REM GETALLEN RADEN
2 PRINT "[CLR]" :REM "[CLR]" IS SHIFT CLR/HOME
5 INPUT "GEEF DE BOVENGRENS"; LI
6 IF LI > 1 THEN 5:REM NIET VALSSPELEN
10 NM = INT(LI*RND(1))+1
15 CN = 0
20 PRINT"IK DENK AAN EEN GETAL."
30 INPUT"WAT IS UW GOK"; GU
35 CN = CN + 1
40 IF GU > NM THEN PRINT"LAGER.....":PRINT:GOTO30
50 IF GU < NM THEN PRINT"HOGER.....":PRINT:GOTO30
60 IF GU = NM THEN PRINT"GOED GERADEN!!!"
65 PRINT"IN"; CN ; "BEURTEN.":PRINT
70 PRINT"NOG EEN KEERTJE? (J/N)";
80 GET AN$:IF AN$ = "J" THEN 2
100 IF AN$ <> "N" THEN 80
110 END
```

De bovengrens van NM kan ingesteld worden. Daarna is het aan u om te raden welk getal de computer in gedachten heeft:

```
GEEF DE BOVENGRENS? 25
IK DENK AAN EEN GETAL.
WAT IS UW GOK? 15
HOGER.....

WAT IS UW GOK? 20
LAGER.....

WAT IS UW GOK? 19
GOED GERADEN!!!
IN 3 BEURTEN.

NOG EEN KEERTJE? (J/N)
```

Een aantal IF/THEN instructies vergelijken uw getal met het door de computer gegenereerde getal. Afhankelijk van uw antwoord zal de computer vertellen of die waarde te hoog of te laag was. Probeert u zelf de formule in regel 10 zo te veranderen, eventueel met gebruikmaking van wat extra regels voor regel 10, dat er ook een ondergrens voor het getal ingegeven kan worden.
Bij elke poging wordt CN opgehoogd. Probeer door goede redenatie het aantal gissingen zo klein mogelijk te houden. Bij ingeven van het juiste getal print de computer een compliment, en het aantal pogingen dat u deed. U kunt dan weer een serie pogingen wagen. Bij elke keer genereert de computer een nieuw willekeurig getal.

Programmeer tips

In de regels 40 en 50 wordt een dubbele punt (:) gebruikt om meer instructies op een regel te scheiden. Dit spaart niet alleen tikwerk, maar ook geheugenruimte, vooral bij langere programma's. Op dezelfde regels lieten we de computer, na IF/THEN iets afdrukken in plaats van direct door te gaan.
Verder ziet u het nut van het nummeren van de regels in stapjes van 10. We kwamen later namelijk pas op de gedachte om het aantal pogingen bij te houden. We konden de daarvoor benodigde regels (15, 35 en 65) eenvoudig tussenvoegen!

Werpt u maar

Het volgende programma simuleert het werpen van 2 dobbelstenen. U kunt er zo mee spelen, of het inbouwen in grotere programma's:

```
5 PRINT "ZULLEN WE WERPEN?"
10 PRINT "RODE DOBBELSTEEN =";INT(6*RND(1))+1
20 PRINT "WITTE DOBBELSTEEN =";INT(6*RND(1))+1
30 PRINT "DRUK OP DE SPATIEBALK...":PRINT
40 GET A$ :IF A$ = "" THEN 40
50 IF A$ = CHR$(32) THEN 10
```

Wilt u het eens proberen? Tracht te begrijpen wat hier gebeurt!

Random graphics

Als laatste oefening met de RND functie, en als inleiding op het maken van tekeningen, geven we u het volgende programmaatje:

```
10 PRINT "[CLR]"
20 PRINT CHR$(205.5+RND(1));
40 GOTO 20
```

Zoals u misschien al zag is regel 20, de regel waar het gebeurt! Nog een functie, de CHR$ (character string) functie, geeft een karakter af bepaald door een standaard code in de reeks "0" ...... "255". Elk van de Commodore 64 karakters is op deze manier gecodeerd. (Zie appendix F) Om snel de code voor een willekeurige karakter te bepalen tikt u in

PRINT ASC ("X")

waarbij X het karakter voorstelt waarvan u de code wenst te bepalen. (Elk van de karakters, dus ook de grafische).
Het antwoord van de computer is de code voor het ingetikte karakter. U begreep natuurlijk al dat ASC weer een functie is: de functie die de standaard ASCII code geeft voor het ingetikte karakter.
Omgekeerd kunt u dat karakter weer afdrukken door het intikken van:

PRINT CHR$ (X) (X is het code getal)

Indien u intikt:

PRINT CHR$ (205); CHR$ (206)

dan ziet u naast elkaar de grafische karakters van de M en de N toets. Dit zijn de 2 karakters die de computer gebruikt voor de opbouw van het doolhof. Met de formule 205.5 + RND (1) genereert de computer een willekeurig getal tussen 205.5 en 206.5. De kans dat het getal boven of onder 206 ligt, is 50%. Omdat CHR$ niet naar breuken kijkt, zal in 50% van de gevallen CHR$ (205) worden afgedrukt, en in de andere 50% van de gevallen CHR$ (206). Als u zin heeft om met dit programma te experimenteren, moet u maar eens een paar tienden van 205.5 aftrekken of erbij optellen. Dit vergroot en verkleint de kans op generatie van CHR$ (205) of CHR$ (206) in evenredige mate!

HOOFDSTUK 5
KLEUR EN GRAPHICS VOOR GEVORDERDEN

HOOFDSTUK 5
KLEUR EN GRAPHICS VOOR GEVORDERDEN

- Kleur en graphics
- Print kleuren
- CHR$ codes van de kleuren
- Peek en Pokes
- Tekenen op het scherm
- Nog meer stuiterende ballen

Kleur en graphics

We hebben tot op dit moment eigenlijk alleen maar aandacht besteed aan de rekenkunst van de Commodore 64. We zijn nog voorbij gegaan aan een van zijn meest fascinerende eigenschappen: het produceren van kleuren en grafische voorstellingen. De stuiterende bal was een simpel voorbeeld, evenals het programma waarmee we een soort doolhof tekenden. Deze voorbeelden laten nog nauwelijks zien welke kracht u eigenlijk bezit. In dit hoofdstuk komen wederom een aantal nieuwe zaken aan de orde. Daarmee leert u het programmeren van kleuren en van grafische voorstellingen, zodat u zelf spelletjes en echte bewegende beelden kunt samenstellen. Omdat we ons tot nog toe beperkt hadden tot het bespreken van rekenvoorbeelden, was het scherm steeds van dezelfde kleur (lichtblauwe tekst op donkerblauwe achtergrond. De rand van het beeld was lichtblauw). In dit hoofdstuk zullen we leren hoe we kleur aan onze programma's kunnen toevoegen. Tevens leren we hoe we al die vreemde grafische tekentjes op de toetsen tot leven kunnen brengen.

Het afdrukken van kleuren

Bij het afregelen van de kleur (hoofdstuk 1) had u al gezien dat de kleuren veranderd konden worden door het indrukken van de <CONTROL> toets samen met een van de kleurtoetsen. Dat werkt prachtig als we direct werken (dus niet in een programma). Hoe krijgen we echter die kleurveranderingen geprogrammeerd? U zag in de listing (= het programma!) van het bal programma dat via PRINT commando's de toetsen voor het bewegen van de cursor geprogrammeerd konden worden. Kleurveranderingen, commando's kunnen gelijk zo in een programma worden ondergebracht. U beschikt over liefst 16 kleuren. Met de <CONTROL> toets, samen met een nummertoets, zijn de volgende kleuren beschikbaar:

1	2	3	4	5	6	7	8
Zwart	Wit	Rood	Cyaan	Paars	Groen	Blauw	Geel

En samen met de Commodore toets:

1	2	3	4	5	6	7	8
Oranje	Bruin	Licht-rood	Grijs	Grijs	Licht-groen	Licht-blauw	grijs

Tik NEW in, en experimenteer een beetje met het volgende:
Druk op de CTRL toets en druk op 1. Druk vervolgens op de K toets zonder op de CTRL toets te drukken. Druk daarna op <CTRL> en 2, laat ze los en druk tenslotte op de LE toets. Werk aldus alle cijfers af, en vorm het woord "kleurig".

10 PRINT "KLEURIG"

RUN
KLEURIG

Net als cursor bewegingen verschijnen ook kleuren als grafische karakters binnen de aanhalingstekens van een PRINT commando. In het bovenstaande voorbeeld verscheen b.v. CTRL 3 als een teken en CTRL 7 als een en teken.
Zo heeft elke kleur zijn eigen teken. In de onderstaande tabel vindt u ze allemaal.

TOETSEN BORD	KLEUR	SCHERM	TOETSEN BORD	KLEUR	SCHERM
CTRL 1	ZWART		C= 1	ORANJE	
CTRL 2	WIT		C= 2	BRUIN	
CTRL 3	ROOD		C= 3	LICHTROOD	
CTRL 4	CYAAN		C= 4	GRIJS 1	
CTRL 5	PAARS		C= 5	GRIJS 2	
CTRL 6	GROEN		C= 6	LICHTGROEN	
CTRL 7	BLAUW		C= 7	LICHTBLAUW	
CTRL 8	GEEL	π	C= 8	GRIJS 3	

Hoewel de print instructie in het programma er nu een beetje vreemd uit zal zien, zal er tijdens het lopen van het programma enkel tekst op het scherm komen. De kleuren van de letters zullen wijzigen conform de door u ingegeven codes. Maak nu zelf wat voorbeelden. Probeer zoveel mogelijk kleur in uw teksten te brengen. Vergeet niet de 8 kleuren te proberen welke u kunt krijgen met de Commodore toets.

TIP:
Na runnen van een programma waarin kleurveranderingen voorkomen, zal het woordje READY en elke volgende tekst verschijnen in de laatst geprogrammeerde kleur. U kunt altijd terug naar de normale kleuren door het indrukken van RUN/STOP en RESTORE.

Kleur CHR$ codes

Kijk even naar Appendix F en lees dan weer verder.

Zoals u zag in de codelijst heeft ook elke kleur, net zoals cursor bewegingen, zijn eigen unieke code. Wij kunnen deze codes gebruiken om hetzelfde resultaat te bereiken als <CTRL> gevolgd door de bijbehorende toets.
Probeer het volgende:

```
NEW
10 PRINTCHR$(147):REM SHIFT CLR
20 PRINTCHR$(30);"CHR$(30) GEEFT KLEUR?"

RUN
CHR$(30) GEEFT KLEUR?

READY.
```

De tekst zou groen moeten zijn. Meestal is het eenvoudiger om de CHR$ functie te gebruiken, vooral wanneer u wilt experimenteren met kleuren. Het volgende programma is een andere manier om een kleurenwaaier te produceren. Omdat er nogal wat gelijke regels zijn, kan regel 40 steeds worden veranderd om de volgende regels te produceren! Dit scheelt nogal wat tikwerk. Na de listing vertellen we u nog eens hoe u dit kunt doen.

```
1 REM DE KLEURENBALKEN
5 PRINT CHR$(147)
10 PRINT CHR$(18); " ";:REM SPATIES
20 CL = INT(8*RND(1))+1
30 ON CL GOTO 40,50,60,70,80,90,100,110
40 PRINT CHR$(5);:GOTO 10
50 PRINT CHR$(28);:GOTO 10
60 PRINT CHR$(30);:GOTO 10
70 PRINT CHR$(31);:GOTO 10
80 PRINT CHR$(144);:GOTO 10
90 PRINT CHR$(156);:GOTO 10
100 PRINT CHR$(158);:GOTO 10
110 PRINT CHR$(159);:GOTO 10
```

Maak regel 5 tot 40 op de gebruikelijke manier. Op het scherm ziet u:

```
1 REM DE KLEURENBALKEN
5 PRINT CHR$(147)
10 PRINT CHR$(18); " ";:REM SPATIES
20 CL = INT(8*RND(1))+1
30 ON CL GOTO 40,50,60,70,80,90,100,110
40 PRINT CHR$(5);:GOTO10
```

Veranderingen aanbrengen (scherm-editing)

Ga met de CRSR naar regel 40, en tik een 5 over de 4 van 40. Loop vervolgens (CURSOR) naar de 5 binnen de haakjes van CHR$. Tik op <SHIFT> en

<INST/DEL> om een extra spatie tussen te voegen, en tik vervolgens 28 en <RETURN>. U ziet nu:

```
1 REM DE KLEURENBALKEN
5 PRINT CHR$(147)
10 PRINT CHR$(18); "     ";:REM SPATIES
20 CL = INT(8*RND(1))+1
30 ON CL GOTO 40,50,60,70,80,90,100,110
50 PRINT CHR$(28);:GOTO 10
```

Geen angst! Regel 40 is er nog. LIST het programma en kijk. Verander nu steeds van de laatste regel het regelnummer en de CHR$ code totdat alle regels van pag. 58 erin zitten. We hadden u toch gezegd dat die verandertoetsen nog eens van pas zouden komen? LIST tenslotte het totale programma om er zeker van te zijn dat het programma compleet is. RUN
Het merendeel van het zojuist ingetikte programma zult u zonder meer begrijpen. In regel 30 daarentegen zit iets nieuws. Maar laten we eerst nagaan wat er precies gebeurt.

In regel 5 wordt de CHR$ code voor CLR/HOME ge"print".
In regel 10 wordt reverse aangezet, en worden er 5 spaties afgedrukt, wat resulteert in een balk, omdat ze immers wit op zwart staan. De eerste keer zal die balk lichtblauw worden, de normale tekstkleur.
Ons werkpaard, de RND functie, verschijnt in regel 20. We kiezen een willekeurige kleur tussen 1 en 8.
Regel 30 bevat een variatie van het IF...THEN commando: ON...GOTO. Een programma kan door middel van ON...GOTO kiezen uit een lijst van regelnummers waar naar toe gesprongen kan worden. Indien de varabele (hier: CL) een waarde van 1 heeft, springt het programma naar het 1e regelnummer uit de lijst (hier: 40). Indien CL=2 zal het 2e nummer gekozen worden, etc.
In de regels 40 ..... 110 vindt een conversie plaats van onze willekeurige getallen naar een bijbehorende CHR$ code. Tenslotte wordt in regel 10 een balkje afgedrukt in die kleur, en begint alles weer van voren af aan.
Probeer zelf uit te vissen hoe u kunt laten kiezen uit 16 willekeurige kleuren. Breid zelf ON...GOTO uit, en voeg regels toe met de overige CHR$ codes.

Peek en poke

Niet dat we direct de computer open willen breken, maar we zouden best eens in zijn geheugen willen kijken of er iets in weg willen stoppen.
Variabelen kunt u zich voorstellen als "doosjes" in de computer waar u uw informatie in heeft weggezet. U kunt zich dan ook voorstellen dat er een aantal bijzondere hokjes kunnen zijn met een voorgeprogrammeerde taak.
De Commodore 64 kijkt naar de inhoud van die hokjes (zogenaamde "geheugenplaatsen") om te zien wat de kleur van het scherm moet zijn, of de kleur van de rand. Ook gebruikt hij geheugenplaatsen om op te slaan welke karakters er op het scherm moeten komen, en nog tal van andere taken.
Door in die geheugenplaatsen andere waarden te plaatsen (POKE) kunnen we de schermkleur veranderen, of afbeeldingen, voorwerpen samenstellen en laten

bewegen. Het is ook de manier waarop we muziek zullen gaan maken. Stelt u zich het volgende voor:

Nummer: | 53280 | 53281 | 53282 | 53283 |

randkleur schermkleur

Op pagina 60 toonden we 4 geheugenplaatsen van de 65536 welke in uw Commodore 64 aanwezig zijn. Plaatsnummer 53280 controleert de kleur van de rand, en plaats 53281 de achtergrondkleur van het scherm.
Probeer:

POKE 53281,7 <RETURN>

Dit zal als gevolg hebben dat de achtergrondkleur van het scherm in geel verandert. 7, de waarde voor geel, is geplaatst in lokatie 53281 welke de achtergrond kleur van het scherm bepaalt. POKE verschillende waarden in 53281 en kijk naar het resultaat. U mag elke waarde tussen de 0 en 255 in het geheugen plaatsen, hoewel alleen de waarden 0 .... 15 zullen werken. De samenhang tussen waarde en kleur is als volgt:

```
10 FOR SC = 0 TO 15
20 FOR RA = 0 TO 15
30 POKE 53280,SC
40 POKE 53281,RA
50 FOR X = 1 TO 2000: NEXT X
60 NEXT RA: NEXT SC
```

Kunt u zelf iets bedenken om de verschillende kleuren zichtbaar te maken? Het volgende voorbeeld is misschien nuttig: We maken gebruik van een tweetal loops om de scherm- en randkleuren te veranderen. De wachtlus in regel 50 vertraagt de afloop. Probeer nu:

PRINT PEEK (53280) AND 15 (RETURN niet vergeten!)

U behoort de waarde 15 op uw scherm te krijgen, de laatste randwaarde welke was toegekend. U ziet in het lijstje van pagina 61 dat dit klopt omdat het scherm nu grijs is. Door het toevoegen van AND 15 elimineert u alle waarden behalve de waarden 0 ... 15. Dit komt door de manier waarop de computer de kleur waarden vasthoudt. Normaal gesproken vindt u met PEEK die waarde welke het laatst door middel van POKE in die lokatie was weggezet. In het algemeen geeft PEEK de waarde die zich op dat moment in de aangegeven geheugenplaats bevindt. Vult u nu zelf het laatste programma aan met PEEK zodat het programma afdrukt wat de waarden zijn van het scherm en de rand.

Oplossing:

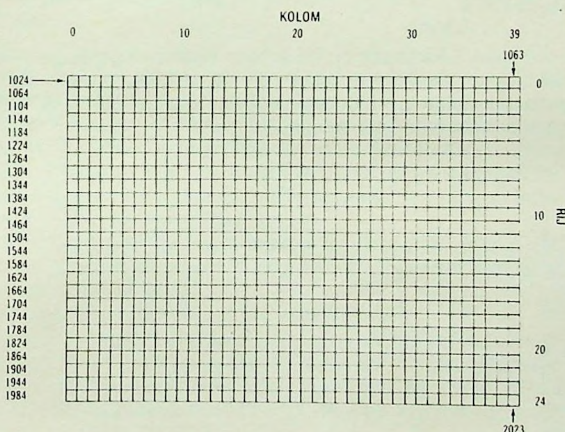
```
25 PRINT CHR$(147);"RAND=";PEEK(53280)AND,15,
"ACHTERGROND="; PEEK(53281)AND15
```

Tekenen op het scherm

Tot nog toe werd alles achter elkaar op het scherm neergezet. Karakter voor karakter werd afgedrukt, al of niet afgewisseld met cursorbewegingen of sprongen naar een volgende tab positie (door middel van de komma) of een nieuwe regel. Om iets op een bepaalde plaats op het scherm te printen, moest u vanaf een bekende positie (meestal HOME) starten, waarna door middel van cursor bewegingen in een printinstructie naar de gewenste plaats gelopen kon worden. Dit neemt echter nogal wat tijd en programmastappen in beslag. Net zoals de schermkleuren bepaald worden door een tweetal lokaties, worden de schermposities en hun inhoud, bepaald door geheugenplaatsen. U kunt daarmee elke gewenste schermpositie "vullen".

De geheugenplaatsen van het scherm

Omdat er op het scherm plaats is voor 1000 karakters (40 posities per 25 regels) ligt het voor de hand dat er 1000 geheugenplaatsen gereserveerd zijn, voor elk van de schermposities een. U kunt zich het scherm voorstellen als 1000 vakjes. Elk vakje correspondeert met een geheugenplaats. In elk van die vakjes past een getal tussen 0 en 255. Er zijn dus 256 mogelijke waarden voor elke geheugenplaats. Deze waarden vindt u terug in appendix E. Door het POKEn van de waarde in de geheugenlokaties behorende bij het schermvakje verschijnt het karakter in die schermpositie.



De geheugenplaatsen van het scherm van de Commodore 64 starten normaliter op nummer 1024 en eindigen dus op 2023. Lokatie 1024 is de linkerbovenhoek van het scherm en zo is 1025 de positie er net naast, etc.... 1063 is de meest rechtse positie van de eerste regel. De geheugenplaats volgend op de laatste positie van een regel is de 1e positie, een regel lager.

Stel dat u nu weer een stuiterende bal wilt programmeren. De bal staat in het midden van het scherm, positie 20 regel van regel 12. Met wat eenvoudig rekenwerk bepalen we de geheugenplaats:

positie op regel

Plaats = 1024 + x + 40*(y) regelnummer (let op: regel 0 is eerste reg!)

De plaats van de bal wordt dus:

1024 + 20 + 480 = 1524

Maak het scherm schoon met SHIFT en CLR/HOME en tik in:

```
POKE 1524,81     karaktercode
POKE 55796,1     kleur
```

geheugenplaats

De geheugenplaatsen voor de kleur informatie

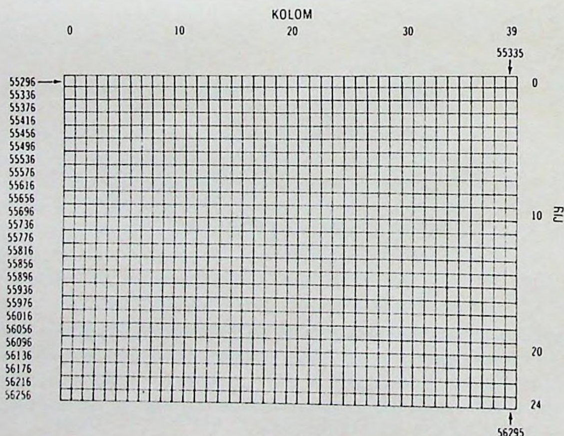
U heeft nu een bal, precies in het midden van het scherm. Zonder gebruik te maken van PRINT heeft u direct in het "schermgeheugen" een bal "gePOKEd". De bal is wit. U kunt ook de kleur van de bal wijzigen middels een andere reeks geheugenplaatsen.

Type: lokatie

```
POKE 55796,2
```

kleur

En de bal wordt rood. Elke positie op het scherm heeft zo 2 geheugenplaatsen. Een ervan bepaalt welk karakter er staat en de andere bepaalt de kleur. De geheugenplaatsen welke de kleuren bepalen, starten op hokje 55296 (links bovenaan) en de volgende 999 hokjes.



We mogen hier dezelfde waarden (0....15) gebruiken als die we gebruikten voor de rand- en achtergrondkleur. Via deze geheugenplaatsen kunnen we rechtstreeks de kleuren bepalen. Onze formule voor het berekenen van de geheugenplaatsen kan eenvoudig aangepast worden voor de kleurlokaties.

Kleur = 55296 + (X) + 40*(Y)

Nog meer stuiterende ballen

Nu volgt een programma waarin we rechtstreeks in het scherm POKEN in plaats van PRINTEN. Met dit programma is het werken veel eenvoudiger. Het is een eerste stap naar het programmeren van ingewikkelder bewegende beelden.

```
10 PRINT "[CLR]"
20 POKE 53280,7 :POKE 53281,13
30 X = 1 :Y = 1
40 DX = 1 : DY = 1
50 POKE 1024+X+40*Y,81
60 FOR T = 1 TO 10 : NEXT
70 POKE 1024 + X + 40*Y,32
80 X = X + DX
90 IF X = 0 OR X = 39 THEN DX = -DX
100 Y = Y + DY
110 IF Y = 0 OR Y = 24 THEN DY = -DY
120 GOTO 50
```

Regel 10 zorgt voor het schoonmaken van het scherm en in regel 20 wordt de achtergrond kleur ingesteld op lichtgroen. De rand van het scherm wordt geel. De variabelen X en Y in regel 30 houden respectievelijk de positie op de regel, en het regelnummer bij. DX en DY in regel 40 bepalen de horizontale en verticale richting van beweging. Steeds als we 1 bij X optellen, gaat de bal naar rechts. Aftrekken van 1 laat de bal naar links bewegen. Evenzo geeft Y + 1 een beweging naar beneden en Y-1 een beweging naar boven.
In regel 50 wordt de bal op het scherm gezet op de positie bepaald door X, Y. De ons al bekende wachtlus in regel 60 zorgt ervoor dat de bal lang genoeg op die lokatie blijft om zichtbaar te zijn.
Regel 70 zet een spatie op de bal: hij verdwijnt weer.
In regel 80 wordt X vergroot of verkleind en regel 90 test of de bal een van beide zijden heeft bereikt. Als dat het geval is, wordt de richting omgekeerd.
Regel 100 en 110 doen hetzelfde voor de boven- en onderzijde. Regel 120 tenslotte stuurt het programma terug naar de weergave van een bal op de nieuwe positie, etc.
De code 81 in regel 50 kan natuurlijk veranderd worden in een andere karaktercode. Dit zal als resultaat hebben dat er in plaats van een bal iets anders over het scherm gaat. Het veranderen van DX of DY in 0 zal tot gevolg hebben dat de bal recht heen en weer gaat in plaats van diagonaal. Het is zelfs mogelijk om een beetje meer intelligentie in het programma in te bouwen. Tot nu toe testten we immers alleen X en Y om te zien of we van het scherm afvlogen. Voeg de volgende regels maar toe:

```
21 FOR L = 1 TO 10
25 POKE 1024 + INT(RND(1)*1000, 166
27 NEXT L
115 IF PEEK(1024 + X + 40*Y) = 166 THEN DX = -DX:GOTO 80
```

De regels 21 tot 27 zorgen ervoor dat er 10 blokken (166 in de CHR$ code voor een blokje) op het scherm worden gezet, op willekeurige plaatsen. In regel 115 wordt getest of onze bal zo'n blok zal raken. Als dat het geval is, wordt de richting van de bal omgedraaid.

HOOFDSTUK 6
SPRITES

HOOFDSTUK 6
SPRITES

- Van Bits en Bytes naar Sprites
- Het opzetten van Sprites
- Meer over Sprites, kleur en beweging

In de voorgaande hoofdstukken zagen we al iets over grafische voorstellingen. De grafische tekens konden gebruikt worden in PRINT instructies. Zo konden we tekeningen maken of beweging in onze schermbeelden krijgen. Een andere methode was middels POKE intructies in de geheugenplaatsen behorende bij de schermposities. Op deze wijze konden we karakters in een keer op de juiste plaats zetten.
Het maken van animaties zou in beide gevallen neerkomen op erg veel werk omdat voorwerpen immers samengesteld zouden moeten worden uit aanwezige bestaande symbolen, en het bewegen ervan vergt veel programmastappen omdat we precies bij moeten houden waar het object zich bevindt. Door de beperking in het gebruik van uitsluitend grafische symbolen zou het gecreëerde voorwerp er ook niet zo uitzien als we graag zouden willen. Het gebruik van Sprites, elimineert veel van deze problemen. Een Sprite is niets anders dan een programmeerbaar tekeningetje dat we elke willekeurige vorm kunnen geven, via BASIC instructies. Het zo gemaakte tekeningetje kan over het hele scherm worden bewogen: het enige dat we daarvoor moeten doen, is het opgeven van de positie op het scherm. De computer doet de rest. Een Sprite kan echter nog veel en veel meer. De kleur kan worden aangepast. U kunt zeggen of de ene Sprite "botst" met de andere. Meerdere Sprites kunnen voor en achter elkaar langs bewegen. U kunt ze vergroten. De moeite die u zich moet getroosten om Sprites te kunnen gebruiken, is minimaal. Voor een goed begrip van de werking van de Sprites moet u namelijk weten hoe een Commodore 64 werkt, en hoe vooral getallen worden verwerkt in de computer. Het is niet al te moeilijk. Wanneer u de voorbeelden navolgt zult u weldra in staat zijn om uw eigen Sprites te maken die de wonderbaarlijke capriolen zullen uithalen.

Van bits en bytes naar sprites

Het tweetallig rekenstelsel

Voor we met Sprites kunnen gaan werken moeten we iets meer weten van hoe de computer intern werkt. Het valt buiten het bestek van dit handboek om erg diep in te gaan op de manier waarop de computer omgaat met getallen. We zullen nochtans proberen u de grondbeginselen te leren welke nodig zijn voor een goed begrip van het werken met ingewikkelde animaties.

BIT

De kleinste hoeveelheid informatie die een computer kan opslaan. Een BIT moet u zich voorstellen als een schakelaar die AAN of UIT staat. Als een BIT "AAN" is, is de waarde 1. Een "UIT" BIT heeft de waarde van 0.

BYTE

Een BYTE is een serie van 8 BITS. U kunt dus 256 verschillende combinaties (of schakelaarstanden, zo u wilt) hebben voor een enkele BYTE.
Als U alle BITS "UIT" heeft, ziet het BYTE er zo uit.

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

de waarde hier is dus 0.

Als alle BITS aanstaan:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

is de waarde 128+64+32+16+8+4+2+1=255

REGISTER

Een REGISTER is een aantal BYTES bij elkaar. In het geval van Sprites is een REGISTER 1 BYTE te lang. Een serie REGISTERS vormen een z.g. REGISTER MAP. Zo'n REGISTERMAP is b.v. het ruitjespatroon dat u gebruikte om een Sprite te tekenen. Elk REGISTER bestuurt een functie zoals b.v. het laten verschijnen van de Sprites (het z.g. ENABLE REGISTER) Het REGISTER dat de Sprite langer en breder maakt, is het X-EXPAND of het Y-EXPAND REGISTER. Onthoud dat een REGISTER een BYTE (een geheugenplaats) is met een specifieke taak.

Binair - decimaal conversie

Decimale waarde

128	64	32	16	8	4	2	1	
0	0	0	0	0	0	0	1	2^0
0	0	0	0	0	0	1	0	2^1
0	0	0	0	0	1	0	0	2^2
0	0	0	0	1	0	0	0	2^3
0	0	0	1	0	0	0	0	2^4
0	0	1	0	0	0	0	0	2^5
0	1	0	0	0	0	0	0	2^6
1	0	0	0	0	0	0	0	2^7

Door het combineren van alle 8 BITS kunnen we elke decimale waarde tussen 0 en 255 voorstellen. Begint u nu te begrijpen waarom de waarden welke we in de geheugenplaatsen POKEn altijd tussen de 0 en 255 liggen? Elke geheugenplaats kan een BYTE informatie bevatten. Elke mogelijke combinatie van 8 nullen en enen geeft een specifieke waarde in het gebied 0...255. Als alle BITS 1 zijn, is de waarde 255. Alles nul geeft de waarde 0. 00000011 is gelijk aan 3 enzovoorts. Dit is het grondbeginsel voor het samenstellen van de gegevens van de Sprites en voor het manipuleren ervan.

We geven nog een voorbeeld van zo'n BYTE:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	1	0	1	0	1	0

en de waarde is 1*128+0*64+1*32+0*16+1*8+0*4+1*2+0*1=170

TIP:

Het volgende programma neemt de conversiezorgen van uw schouders af. We zullen vaak dergelijke omzettingen moeten uitvoeren. Voer het programma in en bewaar het voor later gebruik:

```
5 REM BINAIR NAAR DECIMAAL CONVERSIE
10 INPUT"GEEF BINAIR GETAL (8 BITS):";A$
12 IF LEN(A$) <> 8 THEN PRINT"8 BITS S.V.P..":GOTO10
15 TL = 0: C= 0
20 FOR X=8 TO 1 STEP -1: C=C+1
30 TL = TL + VAL(MID$(A$C,1))*2 ↑ (X-1)
40 NEXT X
50 PRINTA$;" BINAIR = ";TL;"DECIMAAL"
60 GOTO 10
```

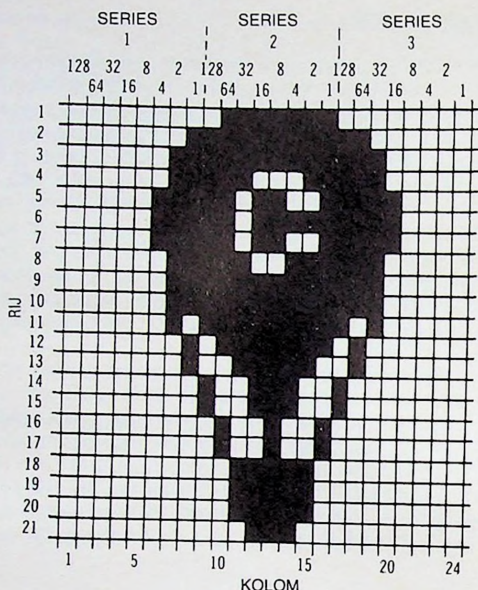
Het programma neemt het binaire getal, dat u in regel 10 als string invoerde en kijkt naar elk karakter (van links naar rechts) d.m.v. de MID$ functie. De variabele C bepaalt met welk van de karakters wordt gewerkt. De VAL functie in regel 30 geeft de waarde van elk karakter. Omdat we nu eenmaal alleen met getallen werken, is de waarde gelijk aan het karakter. Als b.v. het eerste karakter van A$ een 1 is, dan wordt de waarde ook 1. Het laatste gedeelte van 30 vermenigvuldigt de waarde van elk van de karakters met de bijbehorende macht van 2. De eerste waarde (op positie 2^7) geeft aan TL de waarde 1 x 128 of 128. Als dat BIT nul is, wordt de bijbehorende waarde ook nul. Dit proces wordt herhaald voor alle karakters. TL bevat aan het eind de decimale waarde.

Het maken van sprites

Sprites staan onder controle van de beeldbesturing van de Commodore 64. Dit beeld gedeelte bestuurt alles wat er op het scherm verschijnt. Al het moeilijke werk, zoals het weergeven van karakters en grafische tekens, het maken van de kleuren, het laten bewegen, het wordt allemaal door deze beeldbesturing gedaan.
Het stuurgedeelte bevat 46 afzonderlijke AAN/UIT lokaties welke te beschouwen zijn als geheugenplaatsen. Elk van deze lokaties bevat weer een serie van 8 puntjes. Elk puntje kan "AAN" of "UIT" staan. Door het POKEn van een geschikte waarde in zo'n geheugenplaats, kan een Sprite worden aangezet en kan hij worden bewogen. Buiten deze 46 genoemde plaatsen, gebruiken we ook nog een aantal plaatsen uit het grote geheugen van de Commodore 64 om de data op te slaan welke de vorm van de Sprites bepaalt. Tenslotte zijn er nog 8 geheugenplaatsen, direct achter de schermplaatsen, die aangeven waar in het geheugen de computer precies de vorm van de Sprite kan vinden.
Bij het doorwerken van de voorbeelden, zult u merken dat het programmeren rechttoe rechtaan werk is. U zult het snel onder de knie krijgen.
Als u het scherm goed bekijkt, ziet u dat de karakters zijn opgebouwd uit puntjes. Een schermpositie beslaat een blokje van 8 puntjes. Een Sprite nu is 24 van die puntjes lang en 21 puntjes hoog. Elk van de puntjes kan "AAN" zijn (licht) of "UIT" (donker). We kunnen maximaal 8 verschillende Sprites tegelijkertijd laten bewegen. Sprites kunnen bewegen over alle (40 tekens x 8) 320 horizontale puntjes en (25 regels x 8) 200 verticale puntjes van het scherm, onafhankelijk van de inhoud ervan.

Stel dat we een ballon willen maken die we over het scherm willen laten bewegen, dan gaan we als volgt te werk (kijk op pag. 70). De ballon kan getekend worden in de 24 x 21 puntjes welke beschikbaar zijn. De volgende stap bestaat uit het vertalen van die puntjes informatie naar numerieke informatie voor de computer. Dat doen we met ruitjespapier. Teken een rechthoek van 24 x 21 ruitjes en schrijf bij de bovenste horizontale rij resp. de waarden 128, 64, 32, 16, 8, 4, 2, 1. Herhaal dat 2 x zodat er waarden boven elk ruitje staan. Zet langs de linkerkant de waarden 1....21 van boven naar beneden. Schrijf het woordje DATA achter elke regel van 24 hokjes.

Teken nu het plaatje in het raster van 24 x 21 ruitjes. Het is het gemakkelijkst om eerst de contouren te trekken en dan de ruitjes zwart te maken of te kleuren. Stelt u zich nu voor dat elk van de zwart gemaakte ruitjes als puntje op het scherm moet verschijnen. Zo'n ruitje heeft de waarde "1" of is "AAN". Wat donker blijft op het scherm (niet zwart gemaakte ruitjes) is "0" of is "UIT".
Een regel moet nu omgerekend worden, zodat de computer weet wat er staat. We hadden een regel verdeeld in 3 secties van 8 ruitjes, genummerd 128....1. Zo'n sectie is een stuk gegeven dat we een BYTE van 8 hokjes of BITS noemen. De BITS hebben de waarden 127, 64, 32, 16, 8, 4, 2 of 1. De totale waarde voor 8 opeenvolgende blokjes of BITS, de waarde van een BYTE, wordt berekend door de waarde van elk BIT te vermenigvuldigen met "1" als er een puntje moet komen, en met 0 als er geen puntje staat, en de totalen van 8 BITS op te tellen. De eerste regel bevat 3 BYTES. Omdat de eerste 8 hokjes leeg zijn is de waarde van dat BYTE dus 0.



Byte 2 is de middelste sectie van 8 BITS. Hier wordt het resultaat:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

0 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127

De laatste serie puntjes is wederom helemaal "UIT", dus de waarde is 0. Het gegeven (DATA) voor de eerste regel wordt:

DATA 0,. 127, 0

De DATA voor de tweede regel wordt:

SERIE 1	128	64	32	16	8	4	2	1
	0	0	0	0	0	0	0	1

1 = 1

SERIE 2

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255

SERIE 3

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	0

128 + 64 = 192

DATA voor regel 2 wordt derhalve:

DATA 1, 255, 192

Als we zo consequent voortgaan, zullen we uiteindelijk 21 serietjes van 3 getallen hebben, welke de computer aangeven hoe ons object eruit ziet. Ga er even rustig voor zitten.

Hoe nu verder? Tik het nu volgende programma in en kijk wat er gebeurt.

```
1 REM DE GROTE LUCHTBALLON
5 PRINT "[CLR]"
10 V = 53248 :REM START VAN VIDEO CHIP
11 POKE V+21,4:REM ZET Sprite 2 AAN
12 POKE 2042,13 :REM BLOK 13 IS Sprite 2
20 FOR N=0 TO 62:READ Q:POKE832+N,Q:NEXT
30 FOR X=0 TO 200
40 POKE V+4,X :REM VERANDER X COORDINAAT
50 POKE V+5,X :REM VERANDER Y COORDINAAT
60 NEXT X
70 GOTO 30
200 DATA 0,127,0,1,255,192,3,255,224,3,231,224
210 DATA 7,217,240,7,223,240,7,217,240,3,231,224
220 DATA 3,255,224,3,255,224,2,255,160,1,127,64
230 DATA 1,62,64,0,156,128,0,156,128,0,73,0,0,73,0
240 DATA 0,62,0,0,62,0,0,62,0,0,28,0
```

Als u alles netjes heeft overgetikt, zal de ballon zachtjes over het scherm zweven. Om te begrijpen wat er nu precies gebeurt, moeten wij u eerst vertellen welke 46 geheugenplaatsen van invloed zijn op de beweging en positie van de verschillende Sprites. Deze 46 geheugenplaatsen zijn de z.g. REGISTERs van de Sprite besturing.

REGISTER NO	FUNCTIE
0	X coordinaat van de positie van Sprite 0 (op de regel)
1	Y coordinaat van de positie van Sprite 0 (op de regel)
2 - 15	horen steeds 2 aan 2 bij elkaar, voor Sprites 1...7
16	is verdeeld in 8 BITS, hoort bij de X positie
21	is verdeeld in 8 BITS (een voor elke Sprite)
29	ook 8 BITS. Vergroot Sprite in x-richting
23	ook 8 BITS. Vergroot Sprite in y-richting
39 - 46	Kleur van de Sprites 0-7

Bovendien moet u nog weten waar in het geheugen de 64 plaatsen zitten die de vorm van de Sprite bepalen. (Een van die 64 wordt niet gebruikt) Die adressen staan in 8 plaatsen direct achter het schermgeheugen.

Sprite pointer	2040	2041	2042	2043	2044	2045	2046	2047
Sprite no	0	1	2	3	4	5	6	7

De juiste procedure voor het maken en laten bewegen van een Sprite is als volgt:

1. POKE de vorm van de Sprite in het geheugen.
2. Vertel de computer waar in het geheugen de data voor de Sprites staat. Dit doen we door het POKEn van het adres (het nummer van de eerste geheugenplaats) in een van de lokaties 2040 - 2047. (Het adres eerst delen door 64, het adres kan alleen een veelvoud zijn van 64).
3. Laat de Sprites verschijnen door het POKEn van de juiste waarde in REGISTER 21.
4. Verander de X en Y coordinaten om de Sprite te laten bewegen.
5. Eventueel kunt u het voorwerp nog 2 x zo groot maken in X of Y richting (lok. 29 of 23) of u kunt de kleur wijzigen.

UITLEG

In regel 10 staat V = 53248. 53248 Is het eerste adres in het geheugen van de REGISTERs van de beeldbesturing. De juiste geheugenplaats vinden we door het REGISTERnummer op te tellen bij die waarde. Zo zullen 53248 en 53249 respectievelijk de X en de Y positie van de Sprite bevatten.
In regel 11: POKE V + 21,4. POKEn we de waarde 4 in REGISTER 21 (= geheugenplaats 53248 + 21 = 53269). Dit REGISTER bepaalt of een van de Sprites 0....7 aan of uit is (dus weergegeven wordt op het scherm of niet). Deze geheugenplaats of BYTE is weer verdeeld in 8 BITS:

	128	64	32	16	8	4	2	1	
21	0	0	0	0	0	1	0	0	= 4

Als Sprite 2 aan moet zijn, dan zetten we het bijbehorende "bit" op de waarde 1 en komt er dus 1 x 4 als waarde van het BYTE in REGISTER 21. Als u hier dus bij Sprite 3 aan zou willen zetten, zou u de waarde 1 x 8 in 21 moeten zetten, en bijvoorbeeld 3 en 2 aan, zou de waarde 1 x 4 + 1 x 8 = 12 moeten gaan naar REGISTER 21.

In regel 12:

POKE 2042,13

vertelt de computer dat de gegevens voor Sprite nr. 2 (lokatie 2042) staan in het 13e blokje geheugen. Een blokje geheugen is 64 BYTES of geheugen- plaatsen lang. Een Sprite neemt zoals u zag 63 BYTES in beslag, 21 regels van 3 BYTES (die weer opgedeeld waren in 8 BITS)

20 FOR N = 0 to 62: READ Q: POKE 832+N,Q: NEXT

(13 x 64)

In deze regel wordt de eigenlijke Sprite gemaakt. De 63 BYTES data welke de vorm van de Sprite bepalen, worden ingelezen (READ) uit de DATA tabel, en vervolgens in het 13e blokje van 64 BYTES geheugen gePOKEd! Het startadres van die geheugenplaats is 13 x 64 = 832.

30 FOR X = 0 TO 200
40 POKE V+4,X : REM Sprite 2 X-COORDINAAT
50 POKE V+5,X : REM Sprite 2 Y-COORDINAAT

Op school heeft u geleerd dat de X coordinaat de positie bepaalt van een voorwerp langs de (horizontale) X-as en de Y-coordinaat die langs de (verticale) Y-as. Het veranderen van X in de regels 30....50 heeft dus als resultaat dat de Sprite naaar beneden en naar rechts zal bewegen. (De oorsprong of de 0,0 coordinaat bevindt zich links bovenin het scherm!) Elke coordinaat is een puntje van het scherm. Deze positieveranderingen zijn snel genoeg om een indruk te krijgen van vloeiende beweging over het scherm. U vindt meer details in de beschrijving van de REGISTERs in appendix O.

Het is niet mogelijk om via een enkele geheugenplaats de positie van meerdere bewegende voorwerpen te bepalen. Dat is de reden waarom elk van de 8 mogelijk Sprites zijn eigen set X- en Y-registers kent.
Regel 70 zorgt ervoor dat de cyclus weer opnieuw start. Wat er verder nog staat, zijn gegevens voor de tekening. Ziet hij eruit zoals u hem had getekend?
Voeg de volgende regel toe (druk eventueel eerst op RESTORE en RUN/STOP)

25 POKE V+23,4: POKE V+29,4: REM VERGROTEN

en RUN het programma weer. De ballon is nu 2 x zo groot. Door het POKEn van 4 (waarde van het BIT dat Sprite 2 aangeeft) in de REGISTERs 23 en 29 werd ervoor gezorgd dat de Sprite in X en Y richting werd vergroot.

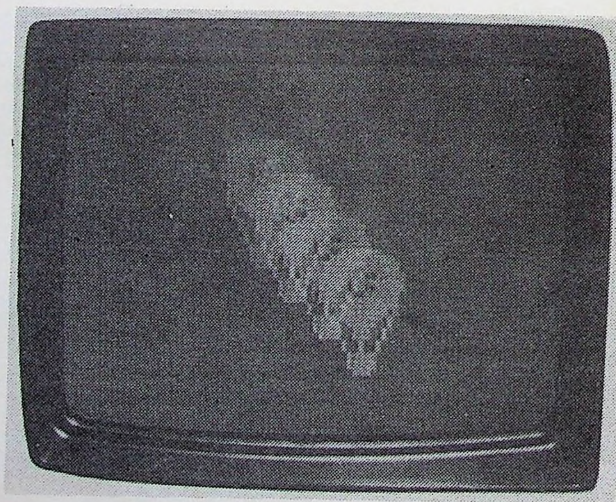
De Sprite startcoordinaat is het punt meest links en boven dat niet dezelfde kleur heeft als de achtergrond. Bij het vergroten blijft die startcoordinaat gelijk. Nog meer opwinding? Verander het volgende:

```
11 POKE V+21,12
12 POKE 2042,13 : POKE 2043,13
30 FOR X = 1 TO 190
45 POKE V+6,X
55 POKE V+7,190-X
```

Nu is er een tweede Sprite welke is aangeschakeld door het POKEn van 12 in het REGISTER dat de Sprites doet verschijnen (nr. 21). De 12 zet zowel Sprite 3 als Sprite 2 aan.
De regels 45 en 55 welke we toevoegden aan het programma, zorgden ervoor dat Sprite 3 bewoog (registers 6 en 7 bepalen resp. X en Y coordinaat van Sprite 3). Nog meer actie in de lucht? Probeer dan:

```
11 POKE V+21,28
12 POKE 2042,13 : POKE 2043,13 : POKE 2044,13
25 POKE V+23,12 : POKE V+29,12
48 POKE V+8,X
58 POKE V+9,100
```

Dit keer hebben we ook nog Sprite 4 aangeschakeld (regel 11). In regel 12 staat dat alle Sprites beschreven worden in hetzelfde geheugenblokje (13) van 64 BYTES. In regel 25 worden de Sprites 2 en 3 groter gemaakt door 12 te POKEn (2 en 3 aan) in het X- en Y-vergrootregister (register nummers 23 en 29).
Regel 48 beweegt Sprite 3 langs de X-as. Regel 58 zet Sprite 3 halverwege het scherm, op positie 100. Omdat deze waarde niet meer verandert, beweegt Sprite 3 alleen horizontaal.



Meer over sprites, kleur en beweging

Na uw eerste experimenten met de Sprites, heeft u recht op wat extra informatie. Om te beginnen kan de kleur van een Sprite ingesteld worden op elk van de 16 verschillende waarden (0...15). Deze waarden vindt u in hoofdstuk 5 of in appendix O.
Zo kan b.v. Sprite 1 licht-groen worden gemaakt door de waarde 13 in REGISTER 40 (dus geheugenplaats 53248+40 = 53288) te zetten:

```
POKE V+40,13 (waarbij V=53248)
```

U zult misschien hebben opgemerkt dat de ballon nooit helemaal naar de rechterzijde van het scherm ging. De reden daarvan is dat de totale X-as 320 puntjes lang is. Een X-register lengte of geheugenplaats kan echter maximaal maar de waarde 255 bevatten. Hoe kunnen we een voorwerp dan toch over het hele scherm laten bewegen? Dat doen we via REGISTER 16 dat tot nog toe niet gebruikt werd. De 8 BITS van dat REGISTER geven aan of de X- lokatie van de bijbehorende Sprite boven of onder de 255 ligt.
In moeilijke bewoordingen zeggen we dat REGISTER 16 de meest significante BITS (de MSB's) bevat van de X-posities van de Sprites. Wanneer een Sprite naar positie 255 is gebracht, wordt het BIT behorende bij die Sprite "1" gemaakt. Voor Sprite 2 zou dat b.v. betekenen:

```
POKE V+16, 4
```

Wanneer we dan naar 256 willen, zetten we 1 in het gebruikelijke X- register. Naar 300? Dan 45 in het gebruikelijke X-register. Omdat we maximaal naar positie 320 kunnen gaan, zal de X-waarde nu maar varieren tussen 0 en 63. Dit allemaal wordt misschien duidelijker aan de hand van de volgende versie van ons oorspronkelijke programma.

```
1 REM DE GROTE LUCHTBALLON
10 V = 53248: POKE V+21,4 :POKE 2042,13
20 FOR N=0 TO 62 : READ Q: POKE 832+N,Q :NEXT
30 FOR X=0 TO 255
40 POKE V+4,X
50 NEXT
60 POKE V+16,4
70 FOR X=0 TO 63
80 POKE V+4,X
90 NEXT
100 POKE V+16,0
110 GOTO 30
```

Regel 60 stelt het MSB in van Sprite 2. Regel 70 verandert de geheugenplaats welke de X-coordinaat bepaalt. Zo beweegt Sprite 2 over het hele scherm. Regel 100 is ook belangrijk. Het MSB van Sprite 2 wordt daar weer afgeschakeld zodat de Sprite weer helemaal links op het scherm komt te staan.

De beste manier om met Sprites te leren werken, is expirimenteren. De Commodore Programmers Reference Guide bevat gedetailleerde informatie over Sprites en de andere Grafische mogelijkheden van uw Commodore 64.

HOOFDSTUK 7
HET MAKEN VAN GELUID

HOOFDSTUK 7
HET MAKEN VAN GELUID

- U bent geen programmeur, toch wilt u geluid.
- Opbouw van een toon – volume
  – golfvorm
  – frequentie
  – omhullende (ADSR)
- Voorbeeld programma's
- Muziek met de Commodore 64
- Een melodie spelen op de Commodore 64
- Geluidseffecten
- Het proberen waard

U bent geen programmeur, en toch wilt u geluid

Geluid, wordt door programmeurs op twee manieren toegepast: nl. voor het maken van muziek en voor het opwekken van geluidseffecten. Voor we de computer muziek kunnen laten spelen moeten we weten hoe een muzieknoot opgebouwd is.

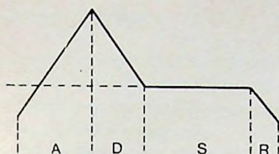
De opbouw van een toon

Er zijn een aantal parameters waarmee een geluid op uw Commodore 64 gedefinieerd kan worden. Het is belangrijk dat u deze kent: volume, golfvorm, attack, decay, sustain, release en frequentie. Deze parameters worden meestal alleen aan het begin ingesteld, door het POKE'n van waarden in bepaalde registers van de Commodore SID geluid-chip. Bepaalde parameters zijn gecombineerd in een register zoals bijvoorbeeld: attack- decay en sustain-release.

De SID chip kent 3 stemmen, die onafhankelijk van elkaar aangestuurd kunnen worden. Dit betekent dat alle parameters voor golfvorm, attack, decay, sustain, release en frequentie driemaal voorkomen.

Voor we het eigenlijke voorbeeld programma bespreken gaan we eerst deze parameters eens nader bekijken.

Wanneer we een muzieknoot spelen dan zien we het volgende plaatje van het geluidsniveau tegen de tijd:



Opmerking: Attack, Decay en Release zijn tijdeenheden. Sustain is een niveau.

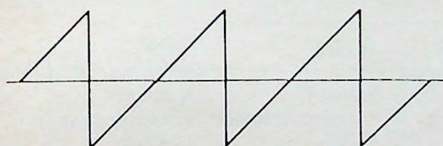
Volume

Het volume kan op 16 verschillende niveaus ingesteld worden via adres 54296. Het grootste volume is 15, geen geluid is 0. Meestal zult u het maximum volume instellen: POKE 54296,15. U zet het geluid af door POKE 54296,0. Het volume register bepaalt het uitgangsniveau van alle drie de stemmen. Het instellen van het volume hoeft slechts eenmaal te gebeuren, en wel aan het begin van het progamma. Het veranderen van het volume tijdens het spelen van een toon, kan interessante effecten geven.

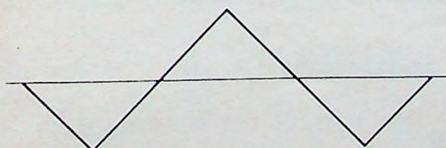
Golfvorm

Met het golfvorm/controle register kunnen we verschillende golfvormen selecteren en de toon starten en stoppen. Het veranderen van de golfvorm heeft tot resultaat dat de klankkleur verandert van bv. een soort xylofoon geluid naar clavecimbel. Elke stem kan zijn eigen golfvorm hebben. De beschikbare golfvormen zijn driehoek, zaagtand, blokgolf en ruis.

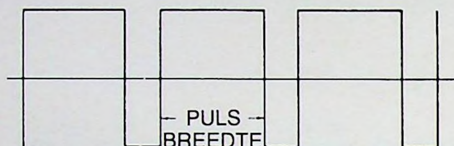
ZAAGTAND



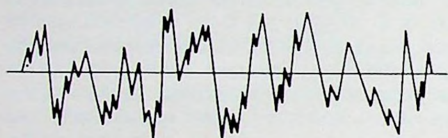
DRIEHOEK



BLOKGOLF



WITTE RUIS (meestal voor geluids efffecten)



GEHEUGEN LOCATIES GOLFVORM REGISTERS

STEM 1	STEM 2	STEM 3
54276	54283	54290

Met het instellen van de golfvorm kan ook de toon gestart worden. De onderstaande tabel laat zien welke waarden in een van de bovenstaande registers gePOKE'd moet worden.

TOON START/STOP WAARDEN

DRIEHOEK	ZAAGTAND	BLOK	RUIS
AAN/UIT	AAN/UIT	AAN/UIT	AAN/UIT
17/16	33/32	65/64	129/128

OPMERKING

Als de blokgolf geselecteerd wordt, moeten ook de pulsbreedte registers van de betreffende stem ingesteld worden.

De navolgende regel start een zaagtand golfvorm in stem 1, en stopt een driehoek-golfvorm in stem 2.

POKE 54276,33: POKE 54283,16

Frequentie

De toonhoogte van een geluid wordt bepaald door de frequentie van de toon die u maakt. Deze frequentie wordt weergegeven door een getal tussen 0 en 65535. Het grootste getal dat we met een POKE instructie kunnen plaatsen is 255, daarom vereist elke noot die u wilt instellen, 2 POKE instructies, een voor de hoge frequentiewaarde (high-byte) en een voor de lage waarde (low-byte).

In appendix M vindt u de POKE waarden voor elke noot uit de 9 octaven die de Commodore 64 kan bestrijken.

Iedere stem heeft haar eigen registerpaar, waarmee de frequentie ingesteld wordt.

STEM	FREQUENTIE	POKE ADRES
1 1	HOOG LAAG	54273 54272
2 2	HOOG LAAG	54280 54279
3 3	HOOG LAAG	54287 54286

Elk van deze drie stemmen kan dus onafhankelijk van de 2 andere worden geprogrammeerd, zodat het programmeren van 3-stemmige muziek of exotische geluidseffecten tot de mogelijkheden behoort.

Om bijvoorbeeld een C (vijfde octaaf) in stem 1 te programmeren moeten de volgende twee POKE's uitgevoerd worden.

```
10 POKE 54273,33                    hoge freq.
15 POKE 54272,135                   lage freq.
```

Voor stem 2 zou het als volgt zijn:

```
10 POKE 54280,33 : POKE 54279,135
```

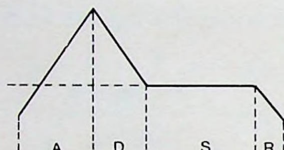
Toets eerst het onderstaand programma in.

```
5 FOR L=54272 TO 54296: POKE L,0: NEXT      maak SID schoon
10 V=54296: W=54276: A=54277                gebruik variabelen om
   S=54278:H=54273:L=54272                  type tijd uit te sparen
20 POKE V,15: POKE A,190: POKE S,89         POKE volume-attack-
                                            decay-sustain-release
30 POKE H,15: POKE L,135                    POKE hi-lo freq.
40 POKE W,33: FOR T=1 TO 200: NEXT          start toon en wacht
50 POKE W,32                                stop toon.
```

Omhullende generator (ADSR)

Als een toon bij een muziekinstrument aangeslagen wordt, neemt de amplitude van 0 toe tot een maximum. De tijd waarin dit gebeurt heet ATTACK, of aanzweltijd. Daarna valt de amplitude terug tot een tussenniveau. De tijd waarin de amplitude van het maximum niveau tot het tussenniveau terugvalt heet DECAY. Het tussenniveau wordt vaak aangeduid met SUSTAIN. Wanneer de toon stopt neemt de amplitude af tot 0. De tijd waarin dit geschiedt heet RELEASE of uitsterftijd.

In het tekeningetje is dit duidelijk te zien.



Met het veranderen van attack, decay, sustain en release is de klankkleur van de toon te beïnvloeden.

Hierna zullen we refereren aan ATTACK/DECAY en het SUSTAIN/RELEASE register. De reden dat we deze parameters gegroepeerd aanduiden is dat deze vier parameters slechts 1/2 byte vragen.

Dit betekent dus dat ATTACK/DECAY net zoals SUSTAIN/RELEASE eerst samengesteld moeten worden en daarna in de overeenkomstige registers moeten worden gePOKE'd.

STEM	TIJD/NIVEAU	POKE ADRES
1	attack/decay	54277
1	sustain/release	54278
2	attack/decay	54284
2	sustain/release	54285
3	attack/decay	54291
3	sustain/release	54292

Attack/decay

De ATTACK maakt gebruik van bit 7-6-5-4, DECAY gebruikt bit 3-2-1-0.

Willen we bijvoorbeeld een hoge attack en een lage decay instellen, tel dan de waarde voor een hoge attack 128 op bij de waarde van de lage decay 2 en POKE het resultaat 130 in het daarvoor bestemde register.

De onderstaande tabel geeft een idee van de mogelijke POKE waarden, die in het ATTACK/DECAY register gebruikt kunnen worden.

HOOG ATTACK	MIDDEL ATTACK	LAAG ATTACK	LAAGSTE ATTACK	HOOG DEC	MIDDEL DEC	LAAG DEC	LAAGSTE DEC
128	64	32	16	8	4	2	1

OPMERKING

De ATTACK tijd kan verhoogd worden door alle ATTACK waarden bij elkaar op te tellen. Bijv : 128 + 64 + 32 + 16 = 240. Op een soortgelijke wijze kan de maximale DECAY bepaald worden door alle DECAY waarden bij elkaar op te tellen. Bijv : 8 + 4 + 2 + 1 = 15. Als u alleen een ATTACK tijd wilt instellen en geen DECAY dan wordt automatisch de DECAY op nul gesteld. Het omgekeerde is ook waar. Wordt bijv. POKE 54277,64 gegeven, dan wordt stem 1 ingesteld op een MIDDEL ATTACK en GEEN DECAY.

Nog enkele voorbeelden.

	STEM	ATTACK	DECAY
POKE 54277,66	1	MIDDEL (64)	LAAG (2)
POKE 54284,100	2	MIDDEL (64) + LAAG (1)	MIDDEL (2)
POKE 54291,15	3	GEEN	MAX (8+4+2+1)
POKE 54291,15	3	MAX (128 + 64 + 32 + 16 + 8 + 4 + 2 + 1)	MAX

Probeer het volgende programma eens, het laat goed de mogelijkheden van ATTACK en DECAY zien.

```
10 FOR L=54272 TO 54296:POKE L,0:NEXT      maak SID chip schoon
20 POKE 54296,15                           volume op maximum
30 POKE 54277,64                           stel attack/decay in
40 POKE 54273,162:POKE 54272,37            POKE een noot in stem 1
50 PRINT"DRUK OP EEN TOETS"                instructie op scherm
60 GET K$: IF K$=""THEN 60                 lees toetsenbord
70 POKE 54276,17: FORT=1TO200:NEXT         start driehoek-golfvorm
80 POKE 54276,16: FORT=1TO50:NEXT          stop toon
90 GOTO 50                                 herhaal
```

Probeer nadat u het programma een aantal malen heeft laten werken, de ATTACK/DECAY instelling te veranderen door regel 30 aan te passen.

```
30 POKE 54277,190
```

U hoort duidelijk het verschil van de toon. Probeert u zelf eens een aantal verschillende waarden te POKEn om een idee te krijgen van de mogelijkheden van ATTACK/DECAY.

Sustain/release

De SUSTAIN/RELEASE registers vertonen qua wijze van programmeren geen verschillen met ATTACK/DECAY. Let op dit betekent niet dat zij dezelfde functie hebben. SUSTAIN is immers een NIVEAU, terwijl attack, decay en release tijden zijn. Het SUSTAIN niveau is instelbaar van 0 tot 15 (het maximale volume), en kan over een onbepaalde tijd aangehouden worden.

HOOG SUSTAIN	MIDDEL SUSTAIN	LAAG SUSTAIN	LAAGSTE SUSTAIN	HOOG REL	MIDDEL REL	LAAG REL	LAAGSTE REL
128	64	32	16	8	4	2	1

OPMERKING

U kunt het SUSTAIN niveau verhogen door alle SUSTAIN waarden bij elkaar op te tellen 128 + 64 + 32 + 16 = 240, welke het MAXIMUM sustain niveau is. Een SUSTAIN niveau van 128 is 50 % van het maximum volume. De SUSTAIN en RELEASE instellingen kunnen op een soortgelijke wijze als ATTACK/DECAY gecombineerd worden, dwz. tel de twee waarden op en POKE het totaal in het register van de overeenkomstige STEM. Om het effect te horen van SUSTAIN voegen we aan het programma de volgende regel toe.

```
35 POKE 54278,128
```

RUN het programma, en beluister het verschil. Met regel 35 vertellen we de computer de toon op 50 % van het maximale volume aan te houden. Door het veranderen 70 kunt u de duur van de toon beïnvloeden. Onthoud dat SUSTAIN de toon aanhoudt op een deel van het maximale volume, als het volume van het maximum terugzakt; het is dus duidelijk geen tijd instelling.

Om het effect te zien van RELEASE veranderen we regel 35 in

```
35 POKE 54278,89 (sustain = 80, release = 9)
```

Probeer ook eens andere waarden.

Voorbeeld programma's

Dit kleine voorbeeld programma laat u nog een keer stap voor stap zien hoe u met de Commodore 64 geluid kunt maken.

1. Kies de stem(men) die u wilt gebruiken, onthoud dat iedere stem haar eigen registers heeft, waarin u de waarden moet POKEn voor attack, decay, etc. U kunt stem 1, 2, 3 tegelijkertijd gebruiken. Het navolgende programma maakt echter alleen gebruik van stem 1.

2. Maak SID chip schoon	5 FORL=54272 TO 54296:POKE L,0:NEXT
3. Volume op maximum	10 POKE 54296,15
4. De aanzweltijd, en de vervaltijd geven aan hoe snel het geluid zijn max. volume bereikt en hoe snel dan het eindvolume bereikt wordt.	20 POKE 54277,190
5. Het nagalmvolume en de uitsterftijd instellen.	30 POKE 54278,248
6. Zoek de frequentie op, of de noot die u wilt weergeven. Deze gegevens staan in de tabel in appendix M. Er moeten 2 waarden worden ingegeven.	40 POKE 54273,17: POKE 54272,37
7. Kies een golfvorm (4 mogelijkheden, 17, 33, 64 of 129)	50 POKE 54276,17
8. Maak een wachtlus, net zo lang als de te spelen noot. (Een kwart noot is + 250 lang. Dit kan echter afwijken omdat lange programma's trager lopen).	60 FORT = 1 TO 250: NEXT
9. En zet de toon uit, door de geselecteerde golfvorm uit te schakelen.	70 POKE 54276,16

Het volgende programma is wat langer, en laat wat meer horen van de muzikale mogelijkheden van uw computer.

```
5 REM TOONLADDER
7 FORL=54272TO54296:POKEL,0:NEXT      maak SID chip schoon
10 POKE 54296,15                      stel volume in
20 POKE 54277,7:POKE54278,133         stel ADSR in
50 READ A                             READ eerste waarde in
55 IF A=-1THEN END                    beëindig loop
60 READ B                             READ tweede waarde in
80 POKE 54273,A:POKE 54272,B          POKE A als Hi-freq en B als
                                      Lo-freq waarde.
85 POKE 54276,17                      start toon
90 FOR T=1TO250:NEXT:POKE54276,16     tijdlus daarna toon uit
95 FORT=1TO50:NEXT                    tijdlus tussen release en
                                      volgende noot
100 GOTO 20                           volgende noot
110 DATA 16,195,18,209,21,31,22,96    getallen uit app. M ieder
120 DATA 25,30,28,49,31,165,33,135    getallenpaar is een noot.
999 DATA -1                           Einde programma (regel 55)
```

Om het timbre van een snaarinstrument te laten horen, verandert u de volgende regels.

```
85 POKE 54276,33
90 FORT=1TO250:NEXT:POKE54276,32
```

En run het programma opnieuw. We hebben de golfvorm veranderd van driehoeksspanning in zaagtand. Dit heeft nogal wat invloed op het timbre. Golfvorm is echter maar een van de verschillende instellingen welke u kunt veranderen om andere geluidseffecten te bekomen. Door het veranderen van de aanzweltijd en het verval is het bijvoorbeeld mogelijk om een geluid te produceren dat lijkt op dat van een banjo. Regel 20 wordt daartoe als volgt veranderd:

```
20 POKE 54277,3: POKE 54278,0         maakt een banjo effect door
                                      geen SUSTAIN te geven
```

Zoals u nu gemerkt heeft, is het mogelijk om de Commodore 64 een geluid te laten produceren dat aan verschillende instrumenten doet denken.

Een melodie spelen met uw Commodore 64

Het volgende programma gebruiken we om een melodie te componeren en af te spelen op onze computer (een stem). Het programma leert ons 2 belangrijke dingen. Let u eens op wat we gedaan hebben met de grote getallen in regel 5. We gebruiken daar de variabele W in plaats van het getal 54276.
Op de tweede plaats is er de manier waarop we met DATA omgaan. Per toon lezen we namelijk steeds 3 waarden in. De hogere frequentie POKE, de lage en de tijdsduur van de noot. In deze melodie tellen we tot 125 voor 1/8 noot of tot 250 voor 1/4, 375 voor drie achtste noot. Tot 500 voor een halve en tot 1000 voor een hele noot. De waarden mogen groter of kleiner genomen worden om zich aan te passen aan het werkelijke tempo, of aan uw eigen smaak. In regel 100 wordt duidelijk hoe de melodie ingevoerd wordt. 34 En 75 zijn de instellingen om noot C te spelen, en het getal 250 is de waarde voor de kwart noot. De eerste toon van onze melodie is dus C, met een duur van 1/4. De tweede noot duurt ook een kwart, maar de toonhoogte is E. Dit gaat zo door tot het eind van de melodie. Op deze manier kunt u zowat elke melodie invoeren. U kunt het aantal DATA regelen, indien nodig uitbreiden. De laatste regel moet zijn DATA-1, -1, -1. Dit is de stop! regel. Door NEW vegen we een vorig programma uit. Tik het volgende programma in en laat de computer spelen.

GOSPELSONG

```
2 FORL=54272TO54296:POKEL,0:NEXT
5 V=54296:W=54276:A=54277:HF=54273:LF=54272:S=54278:
  PH=54275:PL=54274
10 POKEV,15:POKEA,88:POKEPH,15:POKEPL,15:POKES,89
20 READH:IFH=-1THENEND
30 READL
40 READD
60 POKEHF,H:POKELF,L:POKEW,65
80 FORT=1TOD:NEXT:POKEW,64
85 FORT=1TO50:NEXT
90 GOTO10
100 DATA 34,75,250,43,52,250,51,97,375,43,52,125,51,97
105 DATA 250,57,172,250
110 DATA 51,97,500,0,0,125,43,52,250,51,97,250,57,172
115 DATA 1000,51,97,500
120 DATA -1,-1,-1
```

Geluidseffecten

Geluid maken is iets heel anders dan muziek maken. Geluid is verbonden aan bepaalde acties, zoals het gieren van de straaljager of het afschieten van raketten in een of ander ruimtespel. Geluid kan ook simpelweg een piepje zijn om de bediener van een programma te waarschuwen dat hij op het punt staat zijn disk te vernielen. U heeft nogal wat mogelijkheden tot uw beschikking voor het produceren van geluiden. Een tiental suggesties voor het genereren van geluid geeft u misschien ideeën:

1. Het veranderen van de volume-instelling tijdens het spelen van een toon veroorzaakt een echo effect.
2. Tremolo wordt gemaakt door snel de toonhoogte te varieren.
3. Denk eens aan de golfvorm om de aard van het geluid te veranderen.
4. A/V wisseling verandert de snelheid waarmee het geluid op zijn volle volume komt.
5. Nagalmvolume en uitsterftijd. Dit verandert het volume. Interessant wanneer er series van tonen worden geproduceerd.
6. Meerstemmig geluid. U kunt meer dan een enkel geluid tegelijkertijd maken. Elk van die 3 geluiden kan zijn eigen instelwaarde hebben, korter of langer zijn. Een stem kan als echo dienen voor de andere.
7. Natuurlijk kunt u toonhoogten varieren.
8. Met blokgolven met variabele pulsbreedte zijn interessante effecten te bereiken.
9. Witte ruis is het hoofdbestanddeel van slaginstrumenten of geluiden als explosies, schoten of voetstappen. Veranderen van de frequentie instelling geeft een ander type ruis.
10. Probeer snel achtereen de frequentie te varieren over meerdere octaven!

HET PROBEREN WAARD

De volgende programma's kunnen aan bijna elk BASIC programma worden toegevoegd. Wij hopen dat u door deze voorbeelden op programma-ideeën komt. Let eens goed op de manier waarop we het programmeren hebben versneld: het tijdrovend en lastig invullen van de vele POKE adressen hebben we opgelost door de waarden toe te kennen aan variabelen welke gemakkelijk te onthouden zijn. In het voorbeeld hebben we b.v. V voor volume gebruikt, H voor hoge frequentie en L voor lage frequentie. Het programma wordt er begrijpelijker door.

HUILENDE POP

```
5 FORL=54272,TO54296:POKEL,0:NEXT      reset SID-chip
10 S=54272
20 POKES+24,15                         zet volume aan
30 POKES+4,65                          selecteer blokgolf stem 1
40 POKES+5,15                          aanzwel/verval
50 POKES+3,7:POKES+2,128               pulsbreedte op 50%
60 FORX=150TO5STEP-2                   tijdlus
70 POKES+1,40:POKES,X:NEXT             frequentie hoog/laag
80 FORX=150TO5STEP-2                   korte tijdlus
90 POKES+1,40:POKES,X:NEXT             frequentie hoog/laag
100 POKES+4,64                         blokgolf uit.
```

NOOT

De registers voor de pulsbreedte van stem 1 (54274 lo-puls en 54275 hi- puls) worden hier ingesteld op een pulsbreedte van 50 %. Als dit niet gedaan wordt, is de pulsbreedte 0%, waardoor een niet hoorbaar geluid geproduceerd wordt.

SCHIETEN (We maken gebruik van stem 1, witte ruis en teruglopend volume)

```
10 V=54296:W=54276:A=54277:H=54273:L=54272
20 FORX=15TO0STEP-1:POKEV,X:POKEW,129:POKEA,15:POKEH,40:
   POKEL,200:NEXT
30 POKEW,0:POKEA,0
```

HOOFDSTUK 8
DATA MANIPULATIE VOOR GEVORDERDEN

HOOFDSTUK 8
DATA MANIPULATIE VOOR GEVORDERDEN

- Read en data
- Het middelen van getallen
- Geïndiceerde variabelen
- Een-dimensionale matrix
- Nogmaals het gemiddelde
- Het begrip dimensie
- Dobbelsteen met array's
- Twee-Dimensionale array's
- Enquête

Read en data

U hebt kennis gemaakt met twee methoden om waarden toe te kennen aan variabelen. In de eerste plaats was er een directe methode door toewijzen (A = 2 of LET A = 2) en een toewijzen tijdens het lopen van het programma middels de INPUT instructie. Er zijn dikwijls situaties waar geen van beide methoden geschikt is. Met name in het geval dat de waarden dikwijls wijzigen ontstaan er problemen.

Probeert u het volgende programma eens:

```
10 READ X
20 PRINT "X IS NU:"; X
30 GOTO 10
40 DATA 1, 34, 10.5, 16, 234.5
RUN
X IS NU: 1
X IS NU: 34
X IS NU: 10.5
X IS NU: 16
X IS NU: 234.56

?OUT OF DATA ERROR IN 10
READY.
```

In regel 10 leest (READ) de computer een waarde uit de lijst volgend op DATA. Deze waarde wordt toegewezen aan de variabele X. Elke keer dat de lus 10 – 30 – 10

doorlopen wordt, neemt de computer de volgende waarde uit de lijst, kent hem toe aan X en drukt af. De computer onthoudt zelf op welk punt hij gebleven was met het lezen van data. Zo'n verwijzing naar een punt in het geheugen noemt men een pointer (wijzer) of vector.

```
40 DATA 1, 34, 10.5, 16, 234.56
```

Nadat al de waarden gelezen zijn en de computer nog een waarde probeert te vinden, stopt het programma op OUT OF DATA ERROR. Er was immers niets meer te lezen.

- DATA regels hebben een bepaalde vorm.

  ```
  40 DATA 1, 34, 10.5, 16, 234.56
  ```

  — waarden worden door een komma gescheiden.
  — Achter de laatste waarde komt geen komma.

- Achter DATA kunnen getallen staan (hele- of gebroken) of strings. Het is niet mogelijk andere variabelen of bewerkingen in DATA regels te hebben. Het volgende zou dus incorrect zijn:

  ```
  40 DATA A, 23/56, 2*5
  ```

Strings (teksten) kunnen, zoals we al noemden, wel in een DATA regel voorkomen. Het volgende voorbeeld is goed mogelijk:

```
10 FOR X = 1 TO 3
15 READ A$
20 PRINT "A$ IS NU:"; A$
30 NEXT
40 DATA DIT, IS, REUZELEUK

RUN
A$ IS NU: DIT
A$ IS NU: IS
A$ IS NU: REUZELEUK
READY.
```

U ziet dat READ voorkomt in een FOR...NEXT lus. De lus werd net zo vaak doorlopen als het aantal gegevens achter DATA. Deze methode is niet praktisch omdat het aantal gegevens meestal zal wijzigen tijdens het ontwikkelen. Het is zinvoller om als laatste gegeven een z.g. "vlag" neer te zetten: een gegeven met een bekende waarde waarop getest wordt. (Deze waarde moet natuurlijk niet gelijk zijn aan een van de andere gegevens uit de lijst). Zodra deze waarde gelezen wordt, springt het programma door naar het vervolg.

Bijvoorbeeld :

```
10 READ A
15 IF A < 0 THEN END
20 DATA 13,35,29,-999
30 PRINT"GETAL =";A
30 GOTO 10
```

Dezelfde gegevens kunnen, later in het programma, nogmaals gebruikt worden. De instructie RESTORE zorgt ervoor dat de computer weer vooraan begint met het lezen van het eerste gegeven. We illustreren dit door aan het tweede programma van dit hoofdstuk regel 50 toe te voegen.

```
50 GOTO 10
```

U krijgt bij RUN wederom de OUT OF DATA ERROR omdat een hernieuwd uitlezen (na 3 x) niet mogelijk is.

```
45 RESTORE
```

Heft dit probleem op. Nu loopt het programma continu zonder fouten.

Middelen van getallen

Een praktische toepassing van READ en DATA vinden wij bij het inlezen van een aantal getallen en het berekenen van het gemiddelde daarvan:

```
5 T=0 : CT=0
10 READ X
20 IF X=-1 THEN 50 :REM HEBBEN WE ALLES
25 CT=CT+1
30 T=T+X: REM TOTAAL BIJWERKEN
40 GOTO 10
50 PRINT"ER WAREN";CT;"GETALLEN GELEZEN"
60 PRINT"HET TOTAAL WAS";T
70 PRINT"GEMIDDELDE IS"; T/CT
80 DATA 75,80,62,91,87,93,78,-1

RUN
ER WAREN 7 GETALLEN GELEZEN
HET TOTAAL WAS 566
GEMIDDELDE IS 80.8571429

READY.
```

Regel 5 zorgt ervoor dat de variabelen CT (de teller) en T (totaal) gelijk worden aan 0. Regel 10 leest een waarde (READ) en kent die toe aan variabele X. In regel 20 wordt getest of X = -1, hetgeen inhoudt dat we alles hebben ingelezen. Indien dit

nog niet het geval is en de ingelezen waarde deel uitmaakt van de te gebruiken waarden, wordt de teller opgehoogd en wordt X toegevoegd aan het totaal.
Bij het tegenkomen van de -1 vlag springt het programma naar regel 50 waar het aantal ingelezen waarden wordt afgedrukt. Vervolgens wordt in 60 het totaal afgedrukt en tenslotte wordt in regel 70 dit totaal gedeeld door het aantal waarden om tot een gemiddelde te komen.
Door het plaatsen van een DATA vlag, kunt u elk willekeurig aantal getallen achter DATA invullen. Let wel: indien een DATA regel niet voldoende is, kunt u er meerdere maken, al of niet direct op elkaar volgend. Een variatie op het bovenstaande kan het toekennen van waarden zijn aan meer dan een variabele. Achter DATA mogen ook verschillende typen data gemengd voorkomen. Kijk maar naar het volgende voorbeeld waar eerst een naam ingelezen wordt en vervolgens een score van bowlen. Die naam wordt dan afgedrukt samen met de scores en het gemiddelde.

```
10 READ N$,A,B,C
20 PRINT"DE SCORES VAN ";N$;" WAREN:";A;B;C
30 PRINT "ZIJN GEMIDDELDE IS:";(A+B+C)/3
40 PRINT: GOTO 10
50 DATA JAN,190,185,165,EDU,225,245,190
60 DATA LEO,155,185,205,ELS,160,179,187

RUN
DE SCORES VAN JAN WAREN: 190 185 165
ZIJN GEMIDDELDE IS: 180
DE SCORES VAN EDU WAREN: 225 245 190
ZIJN GEMIDDELDE IS: 220

READY.
```

De DATA regels zijn zo samengesteld dat READ elke keer de juiste informatie in een variabele bracht: eerst een naam en vervolgens drie waarden. Anders gezegd: N$ krijgt de eerste keer de waarde "JAN", A wordt 190, B 185 en C gelijk aan 165. Daarna wordt dit proces herhaald met de rest van de informatie (EDU en zijn scores, en LEO en ELS met die van hen).

Geïndiceerde variabelen

Tot op dit moment hebben we alleen gebruik gemaakt van enkelevoudige BASIC variabelen zoals A, A$ of NU. De naam bestond uit een letter gevolgd door nog een letter of een cijfer.
Het valt te betwijfelen dat u meer variabelen nodig zou hebben dan er namen te bedenken zijn door combinatie van letters. Het kan echter zijn dat u omwille van een betere structuur in uw programma, variabelen wilt groeperen. Hiervoor kennen wij de geïndiceerde variabele.

Een geïndiceerde variabele ziet er als volgt uit:

A (0), A (1), A (2), A (3)

variabele index

Een geïndiceerde variabele is een ARRAY (reeks) van variabelen met dezelfde naam, zo'n ARRAY bestaat uit elementen aangeduid door A(1), A(2), A(3) enz. Let op: A, A1 en A(1) zijn 3 verschillende variabelen en wel de enkelvoudige A en A1 en het element A(1) van de geïndiceerde variabele A(N) waarin N de index is voorstelt. Elementen van een reeks van variabelen zijn net als enkelvoudige variabelen de namen van geheugenplaatsen, de dozen met informatie

A(0)	
A(1)	
A(2)	
A(3)	
A(4)	

A(0) is het 1e element. Een computer telt vanaf 0

Als u opschrijft:

```
10 A(0) = 25 : A(3) = 55 : A(4) = 45.3
```

Dan staat in het geheugen:

A(0)	25
A(1)	
A(2)	
A(3)	55
A(4)	45.3

Een ARRAY zoals hier aangeduid is een kolom ARRAY en heeft een dimensie. Later maakt u ook nog kennis met ARRAY's met meer dan een dimensie. De index of de elementaanduiding kan ook samengesteld zijn, of op zijn beurt weer variabelen bevatten. Geldig zijn bv. de indexen:

A(X)   A(X+1)   A(2+1)   A(1*3)

De formules tussen haakjes worden volgens dezelfde regels uitgerekend als wij al in hoofdstuk 2 beschreven.

Nu u de hoofdzaken weet, zult u zich afvragen hoe u er een nuttig gebruik van kunt maken. Een van de toepassingen is het opslaan van een lijst van getallen welke ingelezen zijn via READ of INPUT. We zullen ons voorbeeld van de gemiddelde waarden nog eens uitvoeren, maar nu met gebruikmaking van geïndiceerde variabelen.

```
5 PRINT CHR$(147) :REM SHIFT CLR
10 INPUT "HOEVEEL GETALLEN :";X
20 FOR A=1 TO X
30 PRINT "GEEF WAARDE NO";A;:INPUT B(A)
40 NEXT
50 SOM=0
60 FOR A=1 TO X
70 SOM=SOM + B(A)
80 NEXT
90 PRINT : PRINT "GEMIDDELDE =";SOM/X
99 END

RUN
GEEF WAARDE NO 1 ? 125
GEEF WAARDE NO 2 ? 167
GEEF WAARDE NO 3 ? 189
GEEF WAARDE NO 4 ? 167
GEEF WAARDE NO 5 ? 158
GEMIDDELDE = 161.2

READY.
```

Hoewel het voorbeeld minder omslachtig kon zijn, is het toch een goed voorbeeld van het werken met een ARRAY (of matrix).
In regel 10 wordt er gevraagd hoeveel getallen er ingegeven zullen worden. Deze variabele X bepaalt hoeveel keer er een getal ingegeven zal worden. De ingegeven getallen worden in een ARRAY B geplaatst. De variabele A wordt steeds met 1 opgehoogd, dus elke keer wordt er een ander element van ARRAY B van een waarde voorzien. We zien dus b.v. dat de eerste keer (A=1) de ingegeven waarde geplaatst wordt in geheugenelement B(1), en de tweede keer als A=2, de ingegeven waarde naar B(2) gaat etc. Dit proces gaat door totdat alle waarden ingebracht zijn. Het grote verschil met het eerdere voorbeeld om het gemiddelde te berekenen zit hem hierin dat alle afzonderlijke waarden worden vastgehouden. We kunnen allerlei berekeningen en variaties bedenken met die waarden. In het vorige voorbeeld hielden we enkel een totaal bij van alle ingegeven waarden, zonder dat we in staat waren de afzonderlijke waarden nogmaals te bekijken.
In de regels 50 tot 80 hebben we nog een lus opgezet om de afzonderlijke elementen van de matrix variabele B op te tellen. Daarna drukken we het gemiddelde ervan af. Hier blijkt dat alle waarden inderdaad opgeslagen waren en eventueel terug opgeroepen konden worden.
Als definitief bewijs daarvan laat u alle afzonderlijke waarden nogmaals op het scherm verschijnen:

```
FOR A = 1 TO 5 : ?B(A),: NEXT
```

Op het scherm ziet u de afzonderlijke waarden van de elementen van de matrix B.

Het begrip dimensie

Indien u geprobeerd zou hebben om meer dan 10 getallen in te voeren, zou u DIMENSION ERROR gekregen hebben. We mogen zonder meer ARRAY's gebruiken van 11 elementen (de nummers 0 t/m 10). In een programma gedragen ze zich net zo als de enkelvoudige variabelen. ARRAY's van meer dan 11 elementen moeten "aangegeven" worden. De computer moet van te voren de afmeting, de DIMENSIE, van zo'n set variabelen kennen.

Voeg de volgende regel aan het programma toe:

```
5 DIM B(100)
```

U geeft hiermee aan de computer te kennen dat u ruimte wenst te reserveren voor 101 elementen (vergeet 0 niet).

LET OP: het eerste element van ARRAY B is element B(0) en niet B(1)!

De DIM instructie mag ook gebruikt worden met een variabele. Wanneer u regel 5 weer verwijdert, kunt u dus ook invullen:

```
15 DIM B (X)
```

want in regel 10 geeft u in (= X) hoeveel elementen u in gaat vullen.
Een woord van voorzichtigheid is hier op zijn plaats. Een eenmaal gedimensioneerde ARRAY variabele kan niet nogmaals elders gedimensioneerd worden. Het is wel mogelijk meerdere variabelen op een enkele regel te dimensioneren.

10 DIM C(20), D(50), E(40) bijvoorbeeld.

Dobbelstenen met array's

Het gebruik van ARRAY's geeft in meer complexe programma's het voordeel van minder regels, en eenvoudiger programmeren. Met een enkele ARRAY-variabele kunnen we bijvoorbeeld bij gaan houden hoeveel keer een bepaald aantal ogen is geworpen.

```
1 REM SIMULATIE VAN EEN DOBBELSTEEN
5 PRINT "[CLR]"
10 INPUT "HOEVEEL WORPEN";X
20 FOR L=1 TO X
30 R = INT(6*RND(1))+1
40 W(R) = W(R) + 1
50 NEXT L
60 PRINT"WORP","AANTAL MALEN"
70 FOR C=1 TO 6 :PRINT C, W(C) :NEXT
```

De elementen van variabele W worden gebruikt om bij te houden hoe vaak elk van het aantal mogelijke ogen (1...6) is geworpen. W(2) wordt steeds met één opgehoogd als we 2 werpen, enzovoorts. Door steeds dat element te gebruiken met hetzelfde nummer als het aantal ogen, hebben we ervoor gezorgd dat we niet 6 afzonderlijke variabelen hoefden te benoemen voor het bijhouden van de worpen. Bovendien was het nu ook niet nodig om allerlei tests in te bouwen om te zien wat er geworpen was.
In regel 10 wordt gevraagd hoe vaak men wil simuleren. Regel 20 start het proces van werpen. R is het aantal "geworpen" ogen en in regel 40 wordt element W(R) met één opgehoogd om aan te geven dat dat aantal ogen een keer is geworpen. Als alle worpen gepleegd zijn, drukken regel 60 en 70 een lijstje af van worpen. Op het scherm ziet u iets dergelijks:

```
RUN
HOEVEEL  WORPEN ? 1000
WORP          AANTAL MALEN
 1                160
 2                158
 3                153
 4                189
 5                173
 6                151
```

Dat heeft niet veel moeite gekost nietwaar? Hierna ziet u hoe het voorgaande voorbeeld eruit gezien zou hebben indien we normale variabelen gebruikt zouden hebben. Doet u geen moeite om het in te tikken. De bedoeling is dat u ziet wat er voor extra instructies nodig zijn.

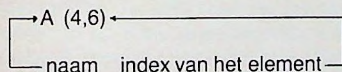
```
10 INPUT "HOEVEEL WORPEN ";X
20 FOR L=1 TO X
30 R = INT(6*RND(1))+1
40 IF R=1 THEN W1=W1 + 1:NEXT
41 IF R=2 THEN W2=W2 + 1:NEXT
42 IF R=3 THEN W3=W3 + 1:NEXT
43 IF R=4 THEN W4=W4 + 1:NEXT
44 IF R=5 THEN W5=W5 + 1:NEXT
45 IF R=6 THEN W6=W6 + 1:NEXT
60 PRINT"WORP","AANTAL MALEN"
70 PRINT 1,W1
71 PRINT 2,W2
72 PRINT 3,W3
73 PRINT 4,W4
74 PRINT 5,W5
75 PRINT 6,W6
```

Dit programma is maar liefst 2 x groter, 16 i.p.v. 8 regels. De ruimtewinst is bij grote programma's nog aanzienlijker, indien we geindiceerde variabelen gebruiken in plaats van enkelvoudige variabelen.

Twee-dimensionale array's

De array-variabelen welke we tot nog toe gebruikten, waren van het een-dimensionale type. Zo'n groep geheugenelementen stelden we ons voor als een verzameling achter elkaar liggende doosjes in het geheugen, waarin zich steeds een element bevond. Hoe zou nu naar uw idee een twee- dimensionale array eruit zien?

Om te beginnen wordt zo'n twee-dimensionale array als volgt omschreven:



We kunnen denken aan een soort rooster structuur:

	0	2	3	4	5	6
0						
1						
2						
3						
4						

De beide indexen moet u maar zien als de nummers van de rij en de kolom van een bepaald element van de matrix. Vullen we de variabele A(3,4) met 255 met de volgende regel:

```
10 LET A(3,4) = 255
```

Dan ziet het array er als volgt uit:

	0	2	3	4	5	6
0						
1						
2						
3				255		
4						

Het toekennen van de waarde 255 aan array element A(3,4) kunnen we ons voorstellen als het plaatsen van die waarde in de 4e kolom van de 3e rij van die verzameling geheugen "dozen". Twee-dimensionale array variabelen volgen dezelfde regels als we al vastgesteld hadden voor een-dimensionale array, dus:

Ze moeten geDIMensioneerd worden	DIM A(20,20)
Het toekennen van een waarde	A (1,1) = 255
Toekennen van de waarde aan een andere variabele	AB = A(1,1)
PRINT de waarde	PRINT A(1,1)

U zou zich kunnen afvragen waarom twee-dimensionale array's, als hun kleinere een-dimensionale broeders al op dezelfde manier werken? Wat zijn er nog meer voor mogelijkheden? Bekijkt u het volgende voorbeeld eens, waar de resultaten van een enquete van 4 vragen met 3 mogelijke antwoorden verwerkt worden.

Enquête

VRAAG 1

BENT U VOOR PERSONAL COMPUTING?

☐ 1 = JA   ☐ 2 = NEE   ☐ 3 = WEET IK NIET

enzovoort

De matrix van antwoorden kunt u zich zo voorstellen:

	Antwoorden		
Vraag 1	ja	nee	blanco
Vraag 2	ja	nee	blanco
Vraag 3	ja	nee	blanco
Vraag 4	ja	nee	blanco

Op de volgende pagina ziet u hoe u de resultaten in kolomvorm op het scherm kunt krijgen.
Veel van de tot nu toe gebruikte programmeertechnieken zijn in dat programma verwerkt. Probeert u zelf de logica uit het programma te halen, ook al heeft u het niet nodig. Het geheel is opgebouwd rond een twee- dimensionale array van 4 x 3 (eigenlijk 5 x 4) elementen, A (3,4). Het totaal aantal antwoorden ja, nee en blanco voor elke vraag wordt bijgehouden in een element.
Om het eenvoudig te houden, gebruiken we de eerste rijen en kolommen van de array (dus A(0,0) tot A(0,4) niet. Bedenk echter steeds dat het eerste element van een array element A(0,0) is en niet A(0,1). Ze zitten er ook altijd in, dus een matrix van 10 elementen is DIM A(9) want A(0) is het eerste element.
Het invullen gebeurt als volgt: als een van de geënqueteerden op vraag 1 "ja" antwoordt, dan wordt element A(1,1) met 1 opgehoogd. Rij 1 voor vraag 1 en kolom 1 voor het antwoord ja. De rest van de antwoorden wordt op gelijkluidende wijze ingevuld. Zo zal een antwoord "nee" op vraag 3 het ophogen met één van element A(3,2) tot gevolg hebben.

```
20 PRINT "[CLR]" :REM "[CLR]" IS SHIFT CLR/HOME
30 FOR R=1 TO 4
40 PRINT "VRAAG NO";R
50 PRINT"1-JA 2-NEE 3-BLANCO"
60 PRINT"WAT WAS HET ANTWOORD :";
61 GET C :IF C<1 OR C>3 THEN 61
65 PRINT C : PRINT
70 A(R,C) = A(R,C) + 1:REM BIJWERKEN EL
80 NEXT R
85 PRINT
90 PRINT"NOG MEER ANTWOORDEN J/N ?";
100 GETA$ :IF A$="" THEN 100
110 IF A$="J" THEN 20
120 IF A$<>"N" THEN 100
130 PRINT "[CLR]";"RESULTATEN:":PRINT
141 PRINT"VRAAG","JA","NEE","BLANCO"
142 PRINT"-----","--","---","------"
143 REM "----" IS [COMMODORE] T
150 FOR R=1 TO 4
160 PRINT R, A(R,1), A(R,2), A(R,3)
170 NEXT R

RUN
VRAAG NO 1
1-JA 2-NEE 3-BLANCO
WAT WAS HET ANTWOORD : 1

VRAAG NO 2
1-JA 2-NEE 3-BLANCO
WAT WAS HET ANTWOORD : 1

enzovoorts...

RESULTATEN:
VRAAG          JA          NEE          BLANCO
----------     -----       -------      -------------
  1            6            1             0
  2            5            2             0
  3            7            0             0
  4            2            4             1
```

AANHANGSELS

AANHANGSELS
VOORWOORD

De steun die Commodore aan zijn gebruikers geeft, houdt niet op na dit eerste gedeelte van het boek waar u een eerste kennismaking opdeed met uw Commodore 64.
Voor het geval dat u het nog niet wist: Commodore bestaat al meer dan 25 jaar. In de zeventiger jaren introduceerden wij 's werelds eerste personal computer: de PET. Vanaf dat moment hebben wij in veel landen van de wereld aan kop gelopen. Omdat wij in staat waren om zelf onze chips te ontwerpen en te fabriceren, konden wij steeds nieuwe en betere microcomputers op de markt brengen tegen een prijs welke altijd veel lager is dan u zou verwachten voor dergelijk technisch vernuft. Commodore ziet het als zijn plicht om niet alleen u, de uiteindelijke gebruiker, te ondersteunen, maar ook de handelaar van wie u de computer kocht. Verder de tijdschriften die artikelen willen publiceren over het hoe en waarom. En last but not least, de software ontwikkelaars die de programma's voor uw computer schrijven. Wij raden u ten zeerste aan om lid te worden van een gebruikersclub. Als lid leert u veel over het gebruik en kunt u ideeën en ontdekkingen uitwisselen.
Er zijn een aantal zeer goede tijdschriften in de handel. Vraag uw handelaar om informatie.
In de volgende aanhangsels vindt u schema's, tabellen en andere informatie die u helpen om uw Commodore 64 sneller en efficienter te programmeren. Ook bevatten ze belangrijke informatie over de vele Commodore produkten die u zouden kunnen interesseren, en een literatuuropgave van zo'n 20 boeken en tijdschriften waarmee u uw talenten verder kunt ontwikkelen en uw kennis up to date kunt houden.

AANHANGSEL A
BESCHIKBARE SOFTWARE

Programmeer hulpmiddelen en computertalen.

C64 101 ASSEMBLER 64

Ontwikkeld voor de ervaren Machinetaalprogrammeur. Het pakket bevat alles dat nodig is om 65xx-serie machinecode programma's te editen, assembleren, laden en te testen. – Makro-Assembler – twee machinetaal monitoren – Editor en loaders – handleiding.

C64 108 – SIMONS BASIC

SIMONS BASIC is de BASIC uitbreiding voor de Commodore 64, 114 extra BASIC kommando's maken het mogelijk om alle mogelijkheden van de Commodore 64 te benutten. SIMONS BASIC bevat o.a. programmeer hulpmiddelen, gestructureerd programmeren, beveiligde input, high-resolution graphics, sprites, muziek, disk-instrukties, getalconversie, extra rekenkundige operatoren en vele instrukties meer.

C64 104 – SUPER-EXPANDER (VSP)

De SUPER-EXPANDER 64 is een krachtige uitbreiding van de BASIC programmeertaal. Vroeger was het nodig om bepaalde geheugen lokaties te PEEK- en POKE'en om de vele mogelijkheden van de Commodore 64 te kunnen benutten. De cartridge biedt U alle mogelijkheden voor graphics, muziek en geluid.

C64 105 – LOGO

LOGO is een edukatieve programmeertaal voor scholieren. Omdat LOGO zo veelzijdig is; eenvoudig voor beginners, maar ook complex genoeg om een uitdaging te vormen voor academici. Een kleine driehoekige cursor, TURTLE genoemd, wordt gebruikt om tekeningen te maken. Deze LOGO bevat naast de standaard instrukties, mogelijkheden zoals : 4 verschillende beeldschermen, 'cataloging' kommando's, een TRACE kommando voor het opsporen van fouten, LIST processing en deze LOGO kent de mogelijkheid om muziek, graphics en tekstbestanden in te lezen welke eerder door LOGO zijn gemaakt.

C64 106 – PILOT

PILOT voor de Commodore 64 is de krachtigste versie van PILOT die verkrijgbaar is. Met dit programma, kunt U uw eigen tekens, kleurrijke verplaatsbare tekeningen (sprites) definiëren, muziek en vele geluidseffecten programmeren. PILOT is een van de beste hulpmiddelen in het onderwijs, onderwijzers kunnen hun eigen programma's schrijven. PILOT is een kombinatie van plezier, ontspanning en leren. Programma's geschreven in de standaard PILOT, kunnen zonder modificaties ingetoetst worden. De PILOT diskette bevat ook een zg. RUN-ONLY versie.

COMAL

COMAL is een gestructureerde BASIC, met extensies die het mogelijk maken om de zg. TURTLE graphics te gebruiken. COMAL wordt erg veel in het onderwijs gebruikt omwille van haar eenvoud. (disk)

FORTH

FORTH 64 is een krachtig operatingsysteem/programmeertaal dat in vele opzichten verschilt van andere programmeertalen. Het is toepasbaar in iedere denkbare zakelijke en procesbesturings toepassing. FORTH 64 is een standaard Fig-FORTH met vele extensies. FORTH 64 kent standaard 400 woorden. (disk)

C64 110 – CP/M 2.2 Operating systeem.

Het CP/M 2.2 operating systeem maakt van uw Commodore 64 een dual-processor personalcomputer. Door het CP/M systeem wordt het aanbod van applicatie programmatuur voor de Commodore 64 vele malen vergroot. Dit eenvoudig te installeren systeem stelt U in staat gebruik te maken van veel toegepaste programma's, zoals tekstverwerking en hogere programmeertalen. (C64 111 Nevada-COBOL en C64112 Nevada-FORTRAN) (CP/M is een geregistreerd handelsmerk van Digital Research, Inc) (diskette + cartridge)

Zakelijke software

CALCRESULT EASY.

Een eenvoudig spreadsheet programma met een dimensie van 64 x 254 cellen

CALC-RESULT ADVANCED

Eindelijk een calculatie programma gemaakt voor de zakenman. Calc-Result maakt al het cijferwerk eenvoudiger. Geschreven voor de wereld van vandaag – gemakkelijk te leren en eenvoudig in het gebruik – Professionele computerkennis is niet vereist om te profiteren van de vele mogelijkheden van Calc-Result. Calc-Result is drie-dimensioneel (64 x 254 x 32) en kent de mogelijkheid tot staafdiagrammen, in kleur.

C64 207 – EASYSCRIPT

Een krachtige tekstverwerker, met groot bedieningsgemak. Verschillende printers kunnen door het programma aangestuurd worden. De vele opties moet U gewoon bij uw dealer gaan bekijken. U kunt daarna niet meer zonder uw eigen tekstverwerker.

C64 208 – EASYSPELL

Foutloos Engels! Dat kan, de met EASYSCRIPT gemaakte teksten kunnen op taalfouten gekontroleerd worden door deze spellingskorrektie module. de module bevat 20000 woorden. Voor hen die een eigen 'vakjargon' (artsen, technici, advokaten etc.) hebben, is het mogelijk een (Nederlands) gebruikers woordenboek

aan te leggen van die woorden die niet in de standaard bibliotheek voorkomen. (Nederlands woordenboek is in voorbereiding – disk).

C64 216 – DE MANAGER

De Manager is een algemene database, gemaakt om de door U ontworpen informatie bestanden te verwerken. De Manager, kan informatie doorgeven aan tekstverwerkers, drukt totalen op het scherm af, maakt speciale bestanden, sorteert op ieder veld en maakt op een eenvoudige wijze lijsten en rapporten. (toetsenbord-disk)

C64 801 – MAGIC DESK I (type en file)
C64 802 – MAGIC DESK II (type en file calc.)

De MAGIC DESK serie programma's zijn een introduktie tot het kantoor van morgen. Op uw computer heeft U de beschikking over een eenvoudige tekstverwerker, rekenmachine, archiefkast, klok, agenda. Met de joystick geeft U Magic Desk de opdrachten. (joystick-disk)

Spel en ontspannings programmatuur.

C64 601 – JUPITER LANDER

Je moet met je ruimtevaartuig voorzichtig op Jupiter zien te landen. Zolang de brandstof nog niet op is, kun je nieuwe landingsplaatsen proberen op te zoeken. Maar je moet wel veilig geland zijn voor de meter in de gele zone komt. Anders ... (joystick/toetsenbord)

C64 602 – KICKMAN

In KICKMAN bestuur je een handige mono-cyclist die punten scoort door ballonnen, geesten en PAC-MAN's te vangen. KICKMAN gebruikt zijn hoofd, voeten, om de vallende objecten te vangen. In sommige fasen van het spel laat KICKMAN de ballonnen springen, om aan extra punten te komen. (joystick/toetsenbord)

C64 603 – SEA WOLF

Als Commandant van een duikboot, moet je vijandelijke schepen tot zinken brengen. Je doelen zijn patrouilleboten, kruisers en aanvalsboten. Je kunt SEA WOLF met een of twee spelers spelen (paddles).

C64 604 – SPEED/BINGO MATH

Twee rekenprogramma's helpen je om je rekenvaardigheid op te voeren, terwijl je nog lol hebt ook. SpeedMath geeft je een tijdslimiet waarbinnen je een varieteit aan rekenkundige vragen op te lossen krijgt. BingoMath vraagt je rekenkundige problemen op te lossen en gebruikt het antwoord om BINGO te spelen. Je kunt tegen de klok of tegen je vrienden spelen. (joystick of toetsenbord)

C64 605 – RADAR RAT RACE

In RADAR RAT RACE, ben je een muis die punten maakt door, in een doolhof, 10 stukjes kaas op te eten. Als je naar de stukjes kaas op zoek bent, word je achtervolgd door ratten ..., en moet je zeker de dodelijke zwarte katten zien te ontlopen. Je hebt drie levens en een beperkte tijd om punten te maken. (joystick-toetsenbord)

C64 606 – CLOWNS

Maak punten met CLOWNS door een hemel van ballonnen te laten springen. Er zijn twee Clowns, een op een wip en de ander zwevend door de lucht. Als de clown naar beneden komt, moet je hem opvangen en de andere clown in de lucht sturen, om de resterende ballonnen lek te prikken. (Paddles)

C64 609 – VISIBLE SOLAR SYSTEMS

Je bent de commandant van een ruimteschip op reis door ons zonnestelsel. Het ruimteschip heeft een bereik van meer dan twee billioen kilometers en is uitgerust met computers die je alles kunnen vertellen over de planeten die je tegen komt. (keyboard)

C64 610 – TOOTH INVADERS

Dit is een spel voor kinderen, de kwade DK verspreidt plak op je tanden. Je moet je tanden poetsen voor ze uitvallen. Als je een tand gepoetst hebt, begint deze te gloeien en verandert van kleur. Je kan DK vernietigen door over hem heen te lopen, maar hij komt altijd terug. Als je er in geslaagd bent alle tanden schoon te maken, begint het fluoride te regenen. (joystick)

C64 613 – LAZARIAN

Je bent de piloot van een ruimte verkenner, gestationneerd in een afgelegen sector van de melkweg. Je opdracht is gestrande 'sterreschepen' te redden en om je sector te verdedigen tegen een scala van bedreigingen. Het spel heeft drie niveau's, en je hebt drie verkenners tot je beschikking. (joystick)

C64 614 – OMEGA RACE

Je bent een Omega-ruimteschip, dat moet wedijveren tegen Droids, de sterkste macht in de melkweg. Je moet de vijandelijke schepen, photon- en dampmijnen vernietigen, die de Droids in de melkweg gelegd hebben. (joystick, paddles, toetsenbord)

C64 616 – LEMANS

Je race-auto is aan de winnende hand in de LeMans Grand Prix. Je moet zoveel mogelijk auto's inhalen om voldoende punten te behalen om in de race te blijven. De wedstrijd gaat over gevaarlijk terrein, gladde en donkere wegen, wegversmallingen en gevaarlijke bochten. Je moet voorzichtig zijn niet te verongelukken als je naar de eindstreep racet. (paddles)

C64 617 - PINBALL SPECTACULAR

PINBALL SPECTACULAR is een Flipperkast op je eigen Commodore 64. Je moet erg snel kunnen reageren. Het spel heeft fantastische geluidseffecten. (paddles)

C64 625 - ZORK I

ZORK I is een spel over het Grote Ondergrondse Rijk; het confronteert je met hachelijke situaties en voorspellingen van mystiek tot macaber. Je doel is het ontdekken van de Twintig Schatten van ZORK I, waarbij de kans groot is dat je voor je leven moet rennen. (Engels/toetsenbord en disk)

C64 626 - ZORK II

Een hulpvaardige Robot, een geslepen eenhoorn, een prinses die om redding vraagt, een demon die al je schatten op eist. Figuren van ZORK II, kunnen niet zoals in ZORK I door zwaarden, bommen en vergif vernietigd worden. (Engels/toetsenbord en disk)

C64 627 - ZORK III

Vreemde en verrukkelijke plaatsen, met oude romeinse aquaducten, dwarrelende mistvelden in het land van de Schaduw. In tegenstelling tot deze feërieke omschrijving, moet je in ZORK III je logisch verstand gebruiken. (Engels/toetsenbord en disk)

C64 628 - DEADLINE

Je bent een hoofdcommissaris die de moord op een schatrijke filantroop moet oplossen. Je hebt 12 uur om de waarheid te ontdekken, voordat het verkeerde testament wordt voorgelezen. Het is een verwarrend mysterie, waarin je snel de draad kwijt raakt. (Engels/toetsenbord en disk)

C64 629 - STARCROSS

STARCROSS is een nieuw science-fiction spel. Je bent een ruimte verkenner in een eenpersoons-ruimteschip op zoek naar zware-gaten. Door geldgebrek moet je je tevredenstellen met een oude betweterige computer en een massa-detector. (Engels/toetsenbord en disk)

C64 630 - SUSPENDED

SUSPENDED speelt zich af op een door een computerbestuurde planeet, dat afstuurt op een gigantische ramp. De computer is zojuist op hol geslagen, je moet de planeet redden door met behulp van zes robots de planeet met de hand te besturen, waarbij je je best moet doen om de oorzaak van de computerstoring te achterhalen. (Engels/toetsenbord en disk)

C64 635 - VOETBAL (INTERNATIONAL SOCCER)

Commodore's VOETBAL spel is één van de meest realistische sport spelen op de Commodore 64. Een spel dat kennis en behendigheid vraagt. VOETBAL kan

gespeeld worden door twee spelers, die ieder een team vertegenwoordigen, maar het is ook mogelijk om tegen de computer spelen, waarbij het niveau van de computer instelbaar is tussen 1 en 9. Het publiek leeft met het spel mee!. (joystick)

MUZIEK

C64 402 – MUSIC MACHINE

De MUSIC MACHINE cartridge maakt van uw Commodore 64 toetsenbord een echte muziek synthesiser. U kunt nu volledig de geluidsmogelijkheden van de Commodore 64 benutten. Zelfs als U niets weet van programmeren of als u geen muziek kunt spelen, 'swingt' MUSIC MACHINE.

C64 403 – MUSIC COMPOSER

U hoeft geen muzikant te zijn om muziek te maken op uw Commodore 64. Deze programma cartridge leert U alles om liedjes en geluidseffecten te maken op uw eigen computer. In de handleiding staan muziekvoorbeelden.

AANHANGSEL B

GEBRUIKSAANWIJZING VOOR UW 1530 DATASETTE

De Commodore 1530 datassette recorder is een apparaat voor het opbergen en terugroepen van computerprogramma's waarbij gewone cassettebandjes gebruikt kunnen worden.

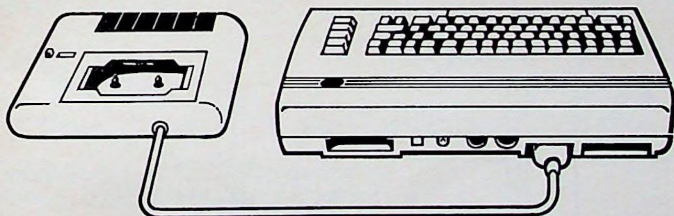
De Commodore 1530 is voozien van een vast verbindingssnoer. Dit snoer verbindt de 1530 met uw computer. De recorder wordt via dit snoer door de computer van spanning voorzien. De communicatie tussen 1530 en de computer geschiedt ook via dit snoer.

SCHAKEL ALTIJD DE COMPUTER UIT ALVORENS DE 1530 OP DE COMPUTER AAN TE SLUITEN!

De 1530 wordt aangesloten op de connector aan de achterkant van de Commodore 64 computer.

De stekker past slechts op een manier op de connector. DUS NIET FORCEREN!

Het is belangrijk om de 1530 op minimaal 60 cm afstand van de TV of monitor te plaatsen, HF-straling kan de juiste werking van de 1530 storen.



Voorlopige controle

Voordat u uw 1530 gaat gebruiken om programma's in te lezen of op te slaan moet u eerst de werking controleren aan de hand van de volgende stappen:

1. Schakel de computer uit, sluit de 1530 aan. (sommige cassette recorders zijn uitgerust met een extra aardings-draadje, deze wordt op de Commodore 64 niet gebruikt. U kunt deze afknippen om te voorkomen dat deze achterin de user-port schiet.)

2. Verzeker u ervan dat de 1530 motor uit is door na te gaan of alle functietoetsen omhoog staan. Is dit niet het geval, dan drukt u op de STOP-toets.

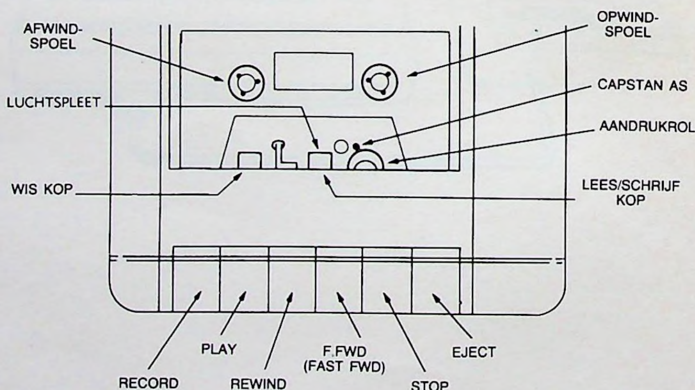
3. Schakel de computer aan.

4. Druk de PLAY-toets van de 1530 in. Kijk of, wanneer de toets wordt ingedrukt, de READ/WRITE-kop naar de spoelen toe beweegt en of de capstan-as in aanraking met komt met de aandrukrol (zie figuur 1). De oprolspoel moet gelijkmatig tegen de klok indraaien.

5. Druk nu op de STOP-toets. De koppen moeten nu teruggaan en de spoel moet stoppen.

6. Druk de REWIND-toets in. De koppen moeten in de rustpositie blijven en de aanvoerspoel moet snel met de wijzers van de klok meedraaien.

7. Druk nogmaals op STOP en vervolgens op F.FWD. De koppen blijven nog steeds in de rustpositie en de oprolspoel moet nu snel tegen de wijzers van de klok indraaien.



8. Druk nogmaals op STOP en probeer dan VOORZICHTIG de RECORD-toets in te drukken. U moet hier een sterke mechanische weerstand ondervinden. NIET FORCEREN.

9. Wanneer alles naar behoren heeft gewerkt, kunt u verder gaan met de controle op de werking op de volgende pagina. Bent u op problemen gestuit, vraagt u dan gerust hulp aan uw dealer.

Controle op de werking

Om de werking van uw nieuwe 1530 te testen moet u een kort BASIC programma schrijven, en dit weg schrijven op cassette (SAVEn). En het hierna weer inlezen (LOADen) in de computer.

Neem een onbespeelde audiocassette (een standaard 1/8″ audiocassette is alles wat u nodig heeft) en plaats deze in de 1530. Druk altijd op REWIND om ervoor te zorgen dat u aan het begin van het bandje bent.

NOOT
Gebruik bandjes met een speeltijd van 0 tot 30 minuten. Gebruik GEEN bandjes met een langere speelduur, deze hebben een te grote mechanische weerstand en belasten de recorder te veel.

Aan de hand van de volgende stappen laten we zien wat er op het beeldscherm gebeurt als u met de cassette recorder werkt.

1. Type in op het toetsenbord:	10 PRINT"DIT IS EEN TEST"
2. Druk in: (RETURN)	
3. Type:	SAVE"TEST"
4. Druk in: (RETURN) Het beeldscherm toont	 PRESS RECORD & PLAY ON TAPE
U doet dit door de RECORD-toets in te drukken totdat beide toetsen vergrendeld zijn.	
Het beeldscherm gaat uit (wordt lichtblauw) en na een tijdje ziet u	OK SAVING TEST READY.
Het programma staat nu op tape Laten we dit controleren	
5. Wis het geheugen met Druk in: (RETURN) Het beelscherm laat zien	NEW READY.

6. Type:	LIST
Druk dan in : (RETURN)	
Het beeldscherm laat zien:	READY.
Dit toont aan dat het geheugen leeg is.	

7. Spoel de cassette terug door REWIND in te drukken, daarna STOP wanneer de band aan het begin is gekomen.

8. Type:	LOAD"TEST"
Het beeldscherm laat zien:	PRESS PLAY ON TAPE
Na het commando te hebben opgevolgd gaat het scherm uit	OK SEARCHING FOR TEST
en kort daarna ziet u	FOUND TEST
na ca. 10 seconden zal de computer het programma automatisch inlezen. Wilt u dat het programma direct ingelezen wordt dan kunt u op de Commodore toets drukken.	LOADING READY

OPMERKING:
Als de LOAD ERROR laat zien, probeer dan het LOAD-commando nog enige malen. Wanneer de fout-melding zich blijft herhalen, lees dan de laatste pagina van dit aanhangsel aandachtig.

9. Type:	LIST
en druk (RETURN). Nu zal het beelscherm laten zien dat het werkgeheugen het programma TEST heeft ingelezen, door het volgende af te beelden:	10 PRINT"DIT IS EEN TEST" READY

Wanneer alle voorgaande stappen succesvol zijn uitgevoerd is uw 1530 op de juiste manier gecontroleerd en klaar voor het gebruik. Alle mogelijke commando's voor het gebruik van uw 1530 zijn op de navolgende pagina's beschreven.

Onderhoud van de bandjes

Spoel altijd de bandjes terug na gebruik, dit beschermt niet alleen de informatie tegen uitwissen maar voorkomt dat de bandjes vervuilen. Bandjes mogen niet in de buurt van sterke magnetische velden, van bijv. speakerboxen of sterke motoren, bewaard of geplaatst worden,

Onderhoud van de 1530

De 1530 gebruikt magneetkopjes om de informatie op cassettebandjes vast te leggen en terug te vinden. Deze LEES/SCHRIJF kopjes hebben de neiging vuil te verzamelen, als het bandje er overheen loopt. Na bepaalde tijd kan deze vuil-ophoping de signaalsterkte verminderen. Het is daarom nodig de volgende procedure na iedere 10 tot 20 bedrijfsuren toe te passen. Hierdoor bent u verzekerd van een goed werkende 1530.

Schoonmaken en demagnetiseren van uw 1530 kopjes

U heeft de volgende hulpmiddelen en materialen nodig:

1. Koppen-reiniger. Alcohol kan worden gebruikt in een noodgeval, maar is niet aan te raden voor langdurig gebruik.

   NOOT

   gebruik in **GEEN** geval **trichlorethaan** of enige ander oplossingsmiddel voor rubber en plastics.

2. Wattenstaafjes.

3. Tape head de-magnitiser. Deze moet een beschermende plastic of rubber bedekking hebben op het gedeelte dat in contact komt met de kopjes. Zodanig dat de kwetsbare luchtspleet niet beschadigd kan worden.

Volg zorgvuldig deze stappen om uw 1530 schoon te maken en te demagnetiseren:

1. Schakel de computer uit.

2. Druk op EJECT om het deksel te openen, druk vervolgens op PLAY om de kopjes bereikbaar te maken.

3. Bevochtig het uiteinde van een wattenstaafje met reinigingsmiddel. Veeg voorzichtig over het oppervlak van het RECORD/PLAY en WISKOPJE (zie fig. 1). Wrijf zachtjes. Iedere opeenhoping van oxide van het bandje op of rond de luchtspleet kan storingen veroorzaken. Maak vervolgens de aandrukrol en de andere bandgeleidingsoppervlakken schoon, mits het reinigingsmiddel hiervoor geschikt is (zie hiervoor het etiket).

4. Sluit de demagnetiseur aan en stel deze in werking. De afstand tussen de demagnetiseur en de kopjes van de 1530 moet dan minimaal 30 cm zijn.

5. Breng de demagnetiseur langzaam naar het RECORD/PLAY-kopje en beweeg deze langzaam rond het oppervlak van het kopje. De snelheid waarmee u dit doet moet ongeveer 2,5 cm per seconde zijn.

6. Breng de demagnetiseur naar het wiskopje en vervolgens naar alle andere metalen oppervlakken die in de nabijheid van de band komen.

7. Haal nu de demagnetiseur weg van de koppen. Stel het magnetisch veld niet buiten werking voordat de demagnetiseur op minimaal 60 cm afstand van de kopjes is.

De procedure om de kopjes schoon te maken en te demagnetiseren is nu compleet. Inspecteer het RECORD/PLAY-kopje op slijtage. Wanneer het bandje een groef heeft gesleten welke enkele banddiktes diep is, kan dit leiden tot problemen bij het inlezen. Is dit het geval, dan is vervanging van het kopje noodzakelijk. Normaal gesproken is dit pas nodig na enige duizendtallen draaiuren.

1530-Werking

Commando's

Hier volgen de commando's welke de 1530 besturen. Tik eenvoudig het gewenste commando, gevolgd door (RETURN). In de hieronder vermelde lijst met commando's staat NAAM voor de naam die u, de gebruiker, aan uw programma gegeven heeft. U moet een naam selecteren die een programma van andere programma's (of databestanden) op eenzelfde bandje doet onderscheiden. De naam dient betekenisvol te zijn voor u. PROG1, PROG2, enz. zijn slechte keuzen, omdat deze moeilijk te onderscheiden zijn van elkaar. Een programmanaam kan niet langer zijn dan 16 karakters.

SAVE"NAAM"

zal een programma wegschrijven door het op te bergen op de 1530.

Voorbeeld SAVE"TEST" zal het programma onder de naam "TEST" wegschrijven.

NOOT

Door alleen SAVE in te tikken wordt het programma zonder naam weg geschreven.

LOAD"NAAM"

zal een programma van band inlezen. Alle overige programma's op het bandje worden overgeslagen.

Voorbeeld: LOAD"TEST" zal het programma TEST van de cassette halen.

Wanneer alleen LOAD wordt ingetikt, wordt het eerste programma dat klaar staat, van het bandje ingelezen.

VERIFY"NAAM"

zal verifiëren of het programma, dat zojuist is weggeschreven, op de juiste wijze is weggeschreven.

Voorbeeld: VERIFY"TEST" zoekt het programma "TEST" op en verifieert het op cassette.

Wanneer de computer antwoordt met	OK READY
dan is het programma op juiste wijze weggeschreven. Wanneer echter de computer antwoord met	VERIFY ERROR OK

is het programma niet op juiste wijze op band gezet. Is dat het geval, dan moet het programma opnieuw weggeschreven worden (met SAVE) en opnieuw geverifieerd worden (met VERIFY). Wanneer dit weer tot een foutmelding leidt, dan werkt de 1530 niet naar behoren, is de cassette van slechte kwaliteit of zijn de magneetkopjes aan een schoonmaakbeurt toe.

NOOT

VERIFY kan ook gebruikt worden voor het doorspoelen van het bandje om een nieuw programma weg te kunnen schrijven (te SAVEn) achter alle eerder weggeschreven programma's op het bandje. De gebruikte methode wordt hieronder beschreven.

Wanneer u klaar bent om een nieuw programma weg te schrijven naar band, moet u het VERIFY"NAAM" commando geven, waarin "NAAM" de programmanaam voorstelt van het programma op band. De computer zal dit laatste programma zoeken en verifieren (met de inhoud van het werkgeheugen), terwijl hij daarbij alle andere programma's overslaat. Het beeldscherm zal de VERIFY ERROR melding tonen, maar de band staat nu precies op het einde van alle eerder opgenomen programma's. Het nieuwe programma kan nu weggeschreven worden (SAVE"NAAM"), waar "NAAM" nu staat voor een unieke naam van het nieuwe programma in het geheugen van de computer. Het nieuwe programma staat nu onmiddellijk achter de andere programma's op band.

Gebruik van de teller

Uw 1530 heeft een teller met drie cijfers, welke dient om weggeschreven (geSAVEd) programma's snel terug te vinden. Bij goed gebruik van de teller kunt u de band vooruit- en terugspoelen naar de plaats van het programma dat u wenst in te lezen (te LOADen). Dit bespaart tijd doordat de computer niet de hele band hoeft af te zoeken.

De teller loopt wanneer de band wordt afgespeeld, opgenomen, vooruitgespoeld of teruggespoeld (respectievelijk PLAY, RECORD, FFWD en REWIND).

Hier volgt de werkwijze om een weggeschreven (geSAVEd) programma snel terug te vinden:

1. Stel vast dat het bandje geheel is teruggespoeld.

2. Zet de teller vervolgens op de nulstand (000) door de zwarte knop naast de teller in te drukken.

3. Noteer de tellerstand wanneer u een programma wilt gaan wegschrijven. Dit nummer geeft aan waar het programma zich op de band bevindt.

4. Spoel het bandje vooruit (FFWD) of terug (REWIND) naar de tellerstand van het gezochte programma.

5. Volg de normale inlees(LOAD)procedure.

NOOT

De teller zal niet accuraat werken als de punten 1 en 2 niet zijn uitgevoerd. Zet de teller niet in de nul-stand als de band niet op het begin staat.

Het werken met bestanden (files)

Ervaren programmeurs zullen verfijndere programma's willen schrijven, waarin grote hoeveelheden gegevens (DATA) verwerkt worden. Deze gegevens kunnen opgeslagen worden in een gegevensbestand (datafile) op cassette. De commando's om gegevensbestanden te kunnen lezen en schrijven, worden in het navolgende besproken.

Gegevensbestanden kunnen ingelezen worden van en weggeschreven worden naar de 1530. Deze kunnen NIET ingelezen worden als een programma, ze kunnen slechts ingelezen worden DOOR een programma. Om met de 1530 te communiceren moet de OPEN-instructie gebruikt worden.

OPEN A,B,C,"NAAM"

Dit commando OPENt een logical file waarbij NAAM het bestand identificeert en

A Een referentienummer van de computer is (LOGICAL FILE),wat tussen 1 en 255 mag liggen. Wanneer uw programma meerdere bestanden tegelijk gebruikt, moet ieder bestand een eigen logical file nummer hebben.

B Moet gelijk zijn aan 1 voor de 1530. Dit is uw apparaatnummer (DEVICE ADDRESS)

C Specificeert of er ingelezen of weggeschreven moet te worden van het bestand volgens:

C=0: lees is van cassetteband

C=1: schrijf naar cassetteband met een END- of FILE-markering wanneer het bestand afgesloten wordt.

C=2: schrijf naar cassetteband met een END- of TAPE-markering wanneer het bestand afgesloten wordt.

Voorbeeld

OPEN 5,1,1,"TEST"

zal via het logical-file 5 een schrijf-bestand openen dat "TEST" heet.

Wanneer aan C een waarde 2 wordt toegekend, wordt een END- of TAPE-markering geschreven aan het einde van het bestand. Wanneer de computer dan opgedragen wordt een bestand te lezen dat zich na "TEST" zou bevinden, zal de computer antwoorden met **FILE NOT FOUND ERROR,** en stoppen. Dit komt omdat het bestand TEST aan de computer vertelt dat de band beëindigd is, ongeacht of dit inderdaad ook het geval is, zodat de computer denkt dat er geen informatie op de band volgt.

C en NAAM hoeven niet gespecificeerd te worden door de gebruiker. Wanneer NAAM ongebruikt blijft, wordt het bestand geopend zonder naam. Wanneer een lees-instructie gegeven wordt aan de computer, leest deze het eerste bestand van de band. Wanneer C niet gespecificeerd is, wordt het bestand geopend voor inlezen.

INPUT#A, D

zal gegevens van cassette halen en ze in de computer verwerken.

A– is het logical filenummer dat gebruikt is in een voorgaande OPEN- instructie welke het inlezen van cassette specificeerde.

D– is de BASIC variabele waaraan de gegevens van band zullen worden doorgegeven. Wanneer woorden ingelezen worden, dan moet D D$ zijn. Als D$ niet gebruikt wordt in dit geval, dan zal de volgende foutmelding worden gegeven:

FILE DATA ERROR.

Voorbeeld:

INPUT#5 ,A$

zal stringdata (woorden) van logical file 5 inlezen. Gegevens worden van cassette ingelezen en toegekend aan BASIC variabele A$.

GET# is een alternatief voor INPUT#. GET# neemt 1 karakter per keer op. GET# leest komma's, dubbele punt enz., terwijl INPUT# dat niet kan.

PRINT#A,D

zal gegevens op cassette wegschrijven,

A– is het logical filenummer dat in een voorgaande OPEN-instructie gebruikt is, waarin het wegschrijven naar band gespecificeerd werd.

D– is de BASIC variabele van waaruit gegevens weggeschreven moeten worden. Als de gegevens uit woorden bestaan moet D$ gebruikt worden.

Voorbeeld

PRINT#5,A$

zal de string A$ via logical file 5 op band wegschrijven, gesteld dat het bestand voor wegschrijven geopend is. In geval het bestand niet op juiste wijze geopend is, zal het beelscherm NOT OUTPUT FILE vermelden.

CLOSE A

zal het logical-file toegekend aan A sluiten, waarbij A het logical filenummer zelf is.

WAARSCHUWING

Wanneer dit commando niet gebruikt wordt nadat alle gegevens zijn weggeschreven, kan het voorkomen dat sommige gegevens niet op cassette weggeschreven staan.
Hieronder volgen enige voorbeeldprogramma's die elk van de bovenstaande commando's gebruiken. U zult merken dat van tijd tot tijd het beeldscherm uitgaat, als de cassette recorder gaat lopen, dit is normaal. Het interne cassette buffer wordt op dat moment overgebracht naar de recorder.

Voorbeeldprogramma 1 - wegschrijven van gegevens:

10 OPEN1,1,1,"TEST FILE"	Open bestand 1 voor wegschrijven met naam TEST FILE.
20 FOR X=1TO100	Voer alles tussen FOR en NEXT 10 maal uit.
30 PRINT#1,X	Schrijf de waarde die variabele X heeft via logical-file 1, weg naar band.
40 NEXT	Ga terug naar regel 20 voor 10 keren.
50 CLOSE1	Sluit het bestand af.

Voorbeeldprogramma 2 - inlezen van gegevens met gebruik van INPUT:

10 OPEN1,1,0"TEST FILE"	Open het bestand voor inlezen uit TEST FILE.
20 INPUT#1,D$:OS=ST	Lees een string in van de band als D$.
30 PRINT D$	Druk de waarde van D$ af op het beeldscherm.
40 IF OS=0 GOTO 20	Controleer de status van de cassette. Als deze OK is ga terug naar 20.
50 CLOSE 1	Sluit het bestand af.

Voorbeeldprogramma 3 - inlezen van gegevens met behulp van GET#:

10 OPEN1,1,0"TEST FILE"	Open het bestand voor inlezen uit TEST FILE.
20 GET#1,D$	Haal 1 karakter van de band naar D$.
25 OS=ST	Stel de EOF variabele veilig in OS (PRINT beïnvloedt immers ST ook).
30 PRINT D$	Druk de karakter in variabele D$ af op het scherm.
40 IF OS=0 GOTO 20	Controleer de status van de cassette. Als deze OK is ga terug naar 20.
50 CLOSE 1	Sluit het bestand af.

De variabele ST is een gereserveerde variabele, welke gebruikt wordt om o.a. het einde van een data bestand te signaleren. De status, welke normaal gesproken de waarde 0 heeft, verandert op het moment dat het laatste gegeven door middel van de INPUT# of GET# is ingelezen. Bij einde bestand (End-Of-File), wordt de waarde van ST = 64.

In de Programmers-Reference Manual wordt dieper op de mogelijkheden van de variabele ST ingegaan.

AANHANGSEL C

COMMODORE 64 BASIC

U heeft in dit handboek de grondbeginselen geleerd van de Basic programmeertaal. Dit was waarschijnlijk voldoende om een begin te maken met programmeren en het taalgebruik. Dit aanhangsel bespreekt alle Commodore Basic commando's en de juiste syntax of vorm. Experimenteer zelf met de commando's. Het is niet mogelijk om uw computer te beschadigen door het intypen van commando's of programma's. Al doende leert men nu eenmaal het best.
Dit aanhangsel is onderverdeeld in 4 gedeelten, en wel

1. **Variabelen en operatoren.** Deze sectie beschrijft de verschillende soorten variabelen, de toegelaten namen ervan en de rekenkundige en logische operatoren.
2. **Commando's.** Hier worden de commando's beschreven, nodig om te werken met programma's en voor het veranderen en wegschrijven ervan.
3. **Instructies.** Beschrijving van de Basic instructies welke gebruikt worden in programma's.
4. **Functies.** Beschrijft de string, numerieke, en PRINT functies.

Variabelen

Er worden 3 soorten variabelen gebruikt binnen de Basic van uw Commodore 64 en wel numerieke variabelen van het floating point (vlottende komma) type en het integere type, en string (alfanumerieke) variabelen. De namen van variabelen beginnen altijd met een letter. Een variabele naam kan uit 2 karakters bestaan. In dat geval mag het tweede karakter een letter of een cijfer zijn. Achter de naam volgt nog het teken % als het om een integer variabele gaat (een heel getal) of een $ in het geval van een string.

Voorbeelden

Floating point variabelen:	A, A5, BZ
Integere variabelen:	A%, A5%, BZ%
String variabelen:	A$, A5$, BZ$

Een ARRAY of MATRIX is niets anders dan een verzameling variabelen met dezelfde naam. De variabelen heten de elementen van de matrix en er worden extra getallen gebruikt om het element nader aan te duiden. Matrices moeten geDIMensioneerd worden met de DIM instructie. Er bestaan floating point matrices zowel als integere en string typen. De naam wordt gevolgd door 2 haakjes waartussen het element nummer wordt aangeduid.

A (7) of BZ%(11), A$(50), PT(20)

OPGELET

3 variabelen mogen niet gebruikt worden. Dit zijn variabelen welke inwendig gebruikt worden door uw Commodore 64. Het zijn de variabelen ST, TI en TI$. ST is een status variabele welke de toestand of status aangeeft bij ingang en uitgang van data. Zo zal de waarde van ST veranderen als er een probleem mocht zijn bij het laden van een programma van disk of tape. De variabelen met naam TI en TI$ hebben betrekking op de ingebouwde klok van de Commodore 64. Variabele TI wordt 60 keer per seconde opgehoogd te beginnen bij nul als de computer wordt aangezet. Hij kan weer op nul worden teruggezet als we TI$ veranderen.
TI$ is een string welke ook steeds door de computer wordt veranderd. De eerste twee karakters bevatten een aantal uren, de middelste 2 het aantal minuten en de laatste twee karakters het aantal seconden dat de computer aan staat. We kunnen de klok "gelijkzetten". Nadien zal de klok steeds de juiste tijd aangeven.

TI$ = "101530" Zet de klok op 10 uur 15 min. en 30 sec.

De klok start steeds op de stand"000000" bij het aanzetten van de computer.

Operatoren

De rekenkundige operatoren welke de Commodore 64 kent zijn:

+ optelling
– aftrekken
* vermenigvuldigen
/ deling
↑ machtsverheffen

Er heerst op een rekenregel een vaste rekenvolgorde. Deze is als volgt: Machtsverheffen, vermenigvuldigen, delen gevolgd door optellen en aftrekken. Deze rekenvolgorde kan eventueel gewijzigd worden door het aanbrengen van haakjes. Berekeningen tussen haken worden altijd eerst uitgevoerd. De operators van gelijk- en ongelijkheid zijn:

= gelijk aan
< kleiner dan
> groter dan
<= kleiner of gelijk aan
>= groter of gelijk aan
<> niet gelijk aan

De Boleaanse operatoren van de Commodore 64 zijn:

AND
OR
NOT

Meestal worden deze operatoren gebruikt om meer voorwaarden binnen een IF...THEN instructie te combineren. Voorbeeld:
IF A=B AND C=D THEN 100 (aan beide voorwaarden moet voldaan worden)

IF A=B OR C=D THEN 100 (aan een van beide voorwaarden moet voldaan worden)

Commando's

CONT (continueren)

Dit commando wordt gebruikt om door te gaan met het uitvoeren van een programma dat gestopt is door het indrukken van de STOP toets of door een STOP of END instructie in het programma. Het programma gaat door vanaf dat breekpunt. CONT werkt niet meer na het aanbrengen van veranderingen aan het programma. CONT werkt ook niet als het stoppen een gevolg is van een fout (ERROR of als u een fout veroorzaakt tijdens de pauze.) In bovenstaande gevallen krijgt u de boodschap "CAN'T CONTINUE ERROR"

LIST

Met LIST kunt u een programma of delen ervan (uit het geheugen) op het scherm zichtbaar maken.

LIST	geeft het hele programma
LIST 10-	geeft regel 10 tot het einde
LIST 10	laat enkel regel 10 zien
LIST -10	laat alles zien vanaf het begin tot en met regel 10
LIST 10-20	geeft de regels met nummers 10 t/m 20 op het scherm.

LOAD

Het LOAD commando is bedoeld om een programma in te lezen in het geheugen. Dit inlezen gebeurt van tape of disk. Het intikken van LOAD gevolgd door RETURN heeft als resultaat dat het eerste het beste programma van tape wordt ingelezen. Het LOAD commando kan gevolgd worden door een naam (max. 15 tekens) tussen aanhaaltekens. Deze naam kan weer gevolgd worden door een komma en een getal of een variabele. Dit getal geeft het nummer aan van het toestel waarin zich het programma bevindt. Geen getal of een 1 is toestel nummer 1, de cassette recorder. De diskdrive heeft als nummer: 8.

LOAD	laadt het volgend programma in vanaf tape.
LOAD"HELLO" of LOAD"HELLO",1	zoekt op tape naar het programma met naam "HELLO" en laadt het in na het vinden ervan.
LOAD A$ of LOAD A$,1	zoekt (en laadt) het programma met de naam welke in A$ zit.
LOAD"HELLO",8	zoekt en laadt het programma "HELLO" vanaf disk.
LOAD"*",8	zoekt en laadt het 1e programma op de disk.

NEW

Dit commando wist elk Basic programma in het geheugen uit en verwijdert alle variabelen. Als u het programma niet weggeschreven heeft met SAVE, bent u het kwijt. **WEES DUS VOORZICHTIG!**
NEW kan ook in een programma worden gebruikt. Wanneer we daar echter aankomen, wordt het programma gewist. Het heeft dus alleen zin als u alles op wilt ruimen na afloop van een programma.

RUN

Met RUN wordt de afloop van een programma gestart bij de 1e regel, dus die met het laagste regelnummer. Een regelnummer achter RUN, dus b.v. RUN 100 laat het programma starten bij die regel:

RUN	programma start bij laagste regelnummer.
RUN 100	programma start bij regel 100.
RUN X	UNDEFINED STATEMENT ERROR. Er mag alleen een bestaand numeriek regelnummer achter RUN staan, en geen variabele.

SAVE

Het SAVE commando zorgt ervoor dat het in het geheugen aanwezige programma weggeschreven gaat worden. SAVE zonder meer, gevolgd door RETURN schrijft weg naar tape. De computer ziet geen kans om te controleren of er soms al iets op die tape staat, dus wees voorzichtig! U zou waardevolle programma's kunnen overschrijven.
SAVE gevolgd door een naam (tussen aanhaaltekens) of een string variabele geeft die naam aan het programma. Op deze manier kunt u programma's uit elkaar houden, en ze sneller terug opzoeken. Na de naam volgt eventueel een komma en een toestelnummer: 1 voor cassette, 8 voor de diskdrive. Er kan bij SAVE na dit toestelnummer nog een tweede nummer komen, 0 of 1. Indien u hier een 1 invult, schrijft de Commodore 64 achter het programma het merkteken "einde van tape". Aan dit teken ziet de computer dat het zinloos is om verder op de tape te zoeken. Indien u toch een LOAD commando op zo'n punt zou geven, antwoordt de computer direct: FILE NOT FOUND ERROR

SAVE:	programma wordt op tape geschreven, zonder naam
SAVE"HELLO"	programma wordt op tape geschreven, naam is "HELLO"
SAVE A$	programma wordt op tape geschreven met naam A$
SAVE"HELLO",8	programma gaat naar disk, met naam "HELLO"
SAVE"HELLO",1,1	programma gaat naar tape, de naam is "HELLO" en achter het programma komt het kenmerk: "Einde tape"

VERIFY

Dit commando vergelijkt het programma in het geheugen met dat op tape. We verifieren hiermee dat het programma inderdaad juist is weggeschreven en dat er niets is misgegaan. (Slechte tape of disk b.v.) VERIFY zonder verdere parameters

vergelijkt het geheugen met het eerstvolgende programma van tape. Verify kan gevolgd worden door een naam en een toestelnummer.

VERIFY	controle tegen het eerste programma op tape.
VERIFY"HELLO"	zoekt "HELLO" op tape, en controleert vervolgens.
VERIFY"HELLO",8	zoekt "HELLO" op disk, en controleert.

Instructies

CLOSE

De CLOSE instructie sluit een bestand of een kanaal dat geopend was via een OPEN instructie. Het getal achter CLOSE is het nummer van het bestand of kanaal dat gesloten moet worden.

CLOSE 2	sluit kanaal #2.

CLR

CLR wist de variabelen in het geheugen, maar laat het programma onveranderd. CLR wordt altijd automatisch gegeven bij RUN.

CMD

CMD routeert de output welke normaal naar het scherm zou gaan (via PRINT, LIST, niet door POKE) naar een ander randapparaat. Dit zou de printer kunnen zijn, maar net zo goed de tape unit of de diskdrive. Het randapparaat moet eerst geOPENd worden. De CMD instructie wordt gevolgd door het referentienummer uit de OPEN instructie.

OPEN 1,4	OPEN verbinding 1 naar randapparaat 4, de printer
CMD 1	alle output gaat nu naar de printer
LIST	Het listen van het programma gebeurt op de printer, en niet op het scherm.

De output komt weer terug op het scherm door:

PRINT#1 : CLOSE 1

DATA

De DATA instructie wordt gevolgd door een lijst van gegevens welke door READ instructies worden gelezen. De gegevens kunnen numeriek zijn, of alfanumeriek. De verschillende items worden gescheiden door komma's. Alfanumerieke gegevens moeten alleen dan tussen aanhaaltekens staan als ze spaties, dubbelpunt, een komma of grafische tekens bevatten. 2 Komma's achter elkaar achter een DATA instructie wordt beschouwd en gelezen als nul, of als een lege string (niets dus).

DATA 12, 14.5, "HALLO,MOE", 3.14

DEF FN

Complexe berekeningen kunnen via deze instructie worden voorzien van een functie aanduiding met een korte naam. Lange formules welke op verschillende plaatsen in een programma gebruikt worden, worden zo eenmaal gespecificeerd, waardoor we veel tijd en vooral ruimte besparen. De functie naam is altijd FN gevolgd door een variabele naam van 1 of 2 karakters. De totale functie moet eerst gedefinieerd worden door DEF gevolgd door een numerieke variabele tussen haakjes. Daarna komt dan de uiteindelijke formule, met de variabele op de juiste plaats. Deze formule kan dan opgeroepen worden waarbij de waarde van de variabele achter de functie aanduiding wordt aangegeven.

```
10 DEF FNA (X) = 12*(34.75 - X/.3)
20 PRINT FNA (7)
```

De waarde 7 wordt ingevuld in plaats van X in de formule.

Het resultaat van dit voorbeeld zou 137 moeten zijn.

DIM (DIMensioneer een matrix)

DIM is noodzakelijk indien er meer dan 11 elementen van een matrix bestaan. Bedenk dat een matrix ruimte in beslag neemt, dus maak hem niet groter dan noodzakelijk is. Het totale aantal elementen van een matrix kunt u uitrekenen door de aantallen elementen in elke dimensie met elkaar te vermenigvuldigen.

```
10 DIM A$ (40), B7(15), CC%(4,4,4)

41 elementen     16 elementen 125 elementen
```

In een DIM instructie kunnen meerdere matrices worden geDIMensioneerd. Een matrix kan in een programma echter maar 1 keer worden gedimensioneerd.

END

Bij het tegenkomen van de END instructie stopt het programma, net zoals bij de laatste regel. Met CONT kunt u eventueel toch verder.

FOR ... TO ... STEP

Deze instructie wordt altijd samen met NEXT gebruikt om een deel van het programma een bepaald aantal malen te herhalen. De juiste vorm is:

FOR (variabele naam) = (start v.d. telling) TO (eind van telling) STEP (variabele naam) = (stapgrootte)
De variabele die gebruikt wordt in de FOR ... NEXT lus wordt steeds veranderd in grootte tijdens het aflopen van het programma. Zonder STEP specificatie is de stapgrootte van die verandering altijd 1. De grenzen van de variabelen liggen tussen de start en eindwaarde van de telling.

```
10 FOR L = 1 TO 10 STEP.1
20 PRINT L
30 NEXT L
```

Zoals u ziet wordt achter de eindwaarde STEP gezet met een stapgrootte. In dit geval wordt er steeds 1 bij de waarde van L opgeteld in plaats van 1. Op deze manier kunt u ook terugtellen.

GET

Met GET kunnen we data van het toetsenbord halen, karakter na karakter. Als GET wordt uitgevoerd wordt het op dat moment ingetikte karakter in de variabele (achter GET) geplaatst. Als er geen toets wordt aangeraakt, wordt er "niets" in de variabele geplaatst of nul.
GET wordt altijd gevolgd door een variabele, meestal een string. Indien we een numerieke variabele gebruiken en we drukken een niet-cijfer toets in, dan stopt het programma met een foutmelding. GET kan in een lus worden geplaatst om te zien of er een toets wordt ingedrukt:

```
10 GET A$: IF A$ = "" THEN 10
```

GET #

De GET# instructie wordt gebruikt met een bestand dat middels OPEN gedefineerd is. Er wordt dan 1 karakter naar binnen gehaald.

```
GET#1, A$:REM HAAL EEN KARAKTER UIT EEN BESTAND
```

GOSUB

GOSUB is te vergelijken met GOTO. Het verschil is dat de computer zich herinnert op welk punt hij bezig was met het programma toen er een GOSUB kwam. Een RETURN instructie zorgt ervoor dat het programma terug springt naar het punt direct na GOSUB. Het gebruik ervan is handig wanneer op meer plaatsen in een programma van een dergelijke routine gebruik gemaakt moet worden. In plaats van het meermalen intypen van dezelfde regels, gebruiken we GOSUB naar dat deel.

```
20 GOSUB 800
```

GOTO of GO TO

De instructie GOTO gevolgd door een regelnummer laat het programma voortgaan bij dat regelnummer.

IF .... THEN

Door middel van IF ... THEN analyseert de computer een bepaalde situatie en aan de hand van het resultaat kan de computer 2 dingen doen. Als het resultaat ja of waar is, wordt de instructie achter THEN uitgevoerd. Dit kan elke Basic instructie zijn. Als het resultaat nee of onwaar is, gaat het programma door met de volgende Basic regel.

De uitdrukking achter IF kan een variabele zijn of een formule. De waarde is ja of waar als het resultaat van de variabele of de formule niet gelijk is aan nul. Meestal zien we in de uitdrukking het gebruik van een van de relationele operatoren (=, <, >, <=, >=, <>, AND, OR, NOT).

```
10 IF X  THEN END
```

INPUT

Middels de INPUT instructie kan een programma gegevens krijgen van de gebruiker. Deze gegevens worden toegevoerd aan een variabele.
Bij INPUT stopt het programma en wordt er een ? op het scherm afgedrukt. De gebruiker kan nu zijn antwoord intikken en RETURN. Een INPUT instructie wordt gevolgd door een variabele naam of een aantal variabele namen, gescheiden door komma's. Tussen INPUT en de eerste variabele mag een boodschap tussen aanhalingstekens voorkomen. Indien er meer dan een variabele is, moeten antwoorden gescheiden worden door komma's. INPUT kan maximaal 80 tekens tegelijkertijd inlezen.

```
10 INPUT "MAG IK UW VOORNAAM"; A$
20 PRINT "GEEF UW CODENUMMER"; : INPUT B
```

INPUT# (spreek uit: inpoetnummer)

INPUT# werkt als INPUT. De data komen nu uit een geOPENd bestand of file.

```
10 INPUT#1, A
```

LET

LET wordt haast nooit gebruikt in programma's: we mogen deze instructie waar nodig, weglaten! Toch draait bijna alles rond deze instructie, omdat de waarde die aan een variabele wordt toegekend, volgt uit:

```
LET A=5 of
LET D$="HALLO"
```

NEXT

NEXT wordt altijd gebruikt samen met de FOR instructie. Bij het bereiken van een NEXT instructie wordt de FOR instructie getest om te zien of de eindwaarde van een lus is bereikt. Als dit niet het geval is, wordt de loopvariabele verhoogd met de waarde welke via STEP is gespecificeerd. Als de lus ten einde is, gaat de uitvoering van het programma door met de eerste instructie volgend op NEXT. NEXT kan gevolgd worden door een variabele naam of een reeks variabele namen, gescheiden door komma's. Zonder specificatie wordt de laatste gestarte loop afgewerkt. Als er variabelen gespecificeerd zijn, worden ze van links naar rechts afgehandeld.

```
10 FOR X=1 TO 100:NEXT
```

ON

ON verandert GOTO en GOSUB commando's in een bijzondere versie van de IF instructie. Achter ON staat een formule welke wordt geëvalueerd. Als het resultaat 1 is, wordt er gesprongen naar het 1e regelnummer uit de lijst achter GOTO of GOSUB. Bij 2 als resultaat springen we naar het 2e regelnummer, enzovoorts. Bij negatief resultaat, nul of een resultaat groter dan het aantal mogelijkheden gaat het programma verder met de eerste instructie na de lijst.

```
10 INPUT X
20 ON X GOTO 10, 20, 30, 40, 50
```

OPEN

De OPEN instructie is de poort naar de buitenwereld, zoals de cassetterecorder, de diskdrive, de printer of zelfs het scherm. OPEN wordt gevolgd door een getal (1...255). Dit is willekeurig en is bedoeld als referentienummer. Er volgt een 2e getal achter het eerste, het zogenaamde devicenummer. Bekende nummers zijn:

1 cassette
2 RS-232C interface (USER-PORT met gebruik van de VC 1011A/B cartridge)
3 scherm
4 printer
8 diskdrive

Na dat 2e getal kan er een derde volgen, ook weer gescheiden door een komma, het zogenaamde secundaire adres. In het geval van de cassetterecorder is dat een 0 voor lezen, 1 voor schrijven en 2 voor schrijven met merkteken "einde van de tape". Bij de diskdrive slaat het 3e getal op het buffer- of kanaalnummer. De printer gebruikt het secundaire adres voor het besturen van functies als bijvoorbeeld "groot printen". In het Reference Manual voor de Commodore 64 vindt u meer bijzonderheden.

10 OPEN 1,3	open een kanaal naar het scherm
20 OPEN 2,1,0,"D"	open een bestand met naam "D" op de cassetterecorder. Er wordt gelezen.
30 OPEN 3,4	open een kanaal naar de printer.
40 OPEN 4,8,15	open het commando kanaal naar de disk.

Zie: CLOSE, CMD, GET#, INPUT# en PRINT#, de variabele ST en Appendix B.

POKE

POKE wordt altijd gevolgd door 2 getallen, variabelen of formules. Het eerste is een geheugenadres; het tweede heeft een waarde tussen 0 en 255. Deze waarde wordt door POKE in de aangegeven geheugenlokatie geplaatst. Een eventueel daar aanwezige waarde wordt overschreven.

PRINT

De PRINT instructie is meestal de eerste instructie welke men leert gebruiken. Er zijn echter een aantal variaties die men goed uit elkaar dient te houden. Achter PRINT kan het volgende voorkomen:

Teksten binnen aanhalingstekens
Variabele namen
Functies
Diverse leestekens zoals ; en ,

De leestekens worden gebruikt om de gegevens op een bepaalde manier te formatteren. Een komma verdeelt het scherm in 4 kolommen; een punt komma onderdrukt spatiering en het gaan naar een nieuwe regel. "PRINT" regels eindigend op komma of puntkomma kunnen gevolgd worden door nieuwe "PRINT" regels alsof het het vervolg is van een voorgaande print instructie.

```
10 PRINT "HALLO"
20 PRINT "HALLO" A$
30 PRINT A+B
40 PRINT J;
60 PRINT A,B,C,D
```

Zie ook de POS, SPC en de TAB functies

PRINT#

Er is weinig verschil tussen deze instructie en PRINT. PRINT# wordt gevolgd door een getal (1...255) dat refereert aan een eerder geopende file. Dit getal wordt gevolgd door een komma en een lijst van de te printen data. Komma's en puntkomma's hebben hetzelfde effect als in PRINT. Bepaalde randapparatuur zal niet reageren op TAB en SPC commando's.

```
100 PRINT#1, "GEGEVENS"; A%,B1,C$
```

READ

READ is de instructie voor het lezen van DATA regels. De gegevens worden toegekend aan de op READ volgende variabelen zodat het programma ze kan benutten. Vermijd het inlezen van strings als READ gevolgd wordt door een numerieke variabele. In dit geval veroorzaken we een TYPE MISMATCH ERROR.

REM(REMark)

REM wordt gebruikt om iets uit te leggen of te verduidelijken. Achter REM kunnen we ook aanvullende aanwijzingen geven. REM instructies beinvloeden het programma op generlei wijze, behalve dan de lengte ervan. Achter REM mag alleen tekst staan. REM is de laatste instructie op een regel.

RESTORE

Wanneer deze instructie wordt uitgevoerd, gaat READ weer terug naar het 1e gegeven uit de lijst(en) volgend op DATA. Op deze manier kunnen gegevens meerdere keren binnen een programma worden gelezen. Achter RESTORE volgen geen parameters.

RETURN

RETURN is onverbrekelijk verbonden met een subroutine waar we heen gestuurd worden door GOSUB. Zodra we in een programma RETURN tegenkomen gaan we terug na de 1e instructie volgend op de GOSUB instructie. Indien we op een andere manier dan via GOSUB bij een RETURN terecht komen, generen we een RETURN WITHOUT GOSUB ERROR.

STOP

De naam zegt het al: met deze instructie stoppen we het programma. Op het scherm verschijnt dan de boodschap BREAK IN xx. xx Is het regelnummer waar de STOP instructie stond. Door middel van CONT kunnen we eventueel verdergaan. STOP wordt gebruikt om de fouten uit programma's te halen.

SYS

SYS wordt gevolgd door een decimaal getal of een numerieke waarde in het gebied 0....65535. Het programma start dan het machinetaalprogramma op dat geheugenadres. SYS is te vergelijken met de USR functie (zie pag. 132) zonder de mogelijkheid van doorgifte van variabelen.

WAIT

WAIT wordt gebruikt om het programma te stoppen, totdat de inhoud van een gegeven geheugenlokatie op een zekere manier veranderd is. WAIT wordt gevolgd door een geheugenplaats en maximaal 2 variabelen.

WAIT X,Y,Z

Als V de inhoud van lokatie X is, dan is de formule die we loslaten:

resultaat=(V ⊕ Z).Y ((V exclusief or Z)and Y)

Als het resultaat nul of vals is, blijft het programma de waarde van X testen. In het geval de vergelijking waar is, gaat het programma verder.

Numerieke functies

ABS(X) (absolute waarde)

ABS geeft de absolute waarde van het getal X, dus het getal zonder + of - teken. Het antwoord is altijd positief.

ATN(X) (arctangens)

ATN geeft de hoek, in radialen, van de hoek met tangens gelijk (X)

COS(X) (cosinus)

Geeft de waarde van de cosinus van hoek (X). (X) is in radialen.

EXP(X)

Geeft de waarde van het getal e (2.71827183) tot de macht (X).

FN xx(X)

Geeft de waarde van de functie xx welke door de gebruiker via een DEF FNxx(XX) instructie werd gecreëerd.

INT(X)

Geeft de integere waarde van X, dat is het hele getal kleiner of gelijk aan (X). Bijvoorbeeld INT(1.26)= 1 en INT(-7.896)=-8

LOG(X)

Geeft de natuurlijke logaritme van (X). De log met grondtal 10 krijgt men door het resultaat te delen door log(10). $^{10}\log(X) = \ln(X)/\ln(10))$

PEEK(X)

Geeft de waarde van geheugenplaats (X). X kan liggen tussen 0 en 65535. Het resultaat ligt tussen 0-255. PEEK wordt vaak samen met POKE gebruikt.

RND(X)

RND(X) geeft een willekeurig getal in het gebied 0-1. (X) maken we gelijk aan 1 of een of ander positief getal. Als X nul is, krijgen we hetzelfde getal als de vorige keer. RND(RND(1)) verzekert ons van een onbekende start. Door een negatief getal wordt het proces opnieuw gestart. Tip voor valsspelers! Door het gebruik van steeds hetzelfde negatieve getal, krijgen we steeds dezelfde reeks willekeurige getallen.
De formule voor het genereren van een getal tussen de grenzen X en Y is:

N = RND(1)*(Y-X)+X

waarin Y de bovengrens is en X de ondergrens van het gewenste gebied.

SGN(X)

Uit deze functie komt het teken van X (+, – of 0). Het resultaat is +1 als X positief is, -1 als X negatief is en 0 als X nul is.

SIN(X) (sinus)

SIN(X) is de trigonometrische sinusfunctie. Het resultaat is de sinus van X met X in radialen.

SQR(X) (vierkantswortel)

Geeft de vierkantswortel uit X, waarbij X>=0. Indien X<0, krijgen we een ILLEGAL QUANTITY ERROR.

TAN(X) (tangens)

Geeft de tangens van hoek X, tg(X) waarbij X weer in radialen.

USR(X)

Bij gebruik van deze functie springt het programma naar een programma in machinetaal. Het startpunt van die routine staat in 2 geheugenplaatsen aangegeven. De parameter (X) wordt doorgegeven aan het machinetaal- programma. Na afloop komt een resultaat in een (andere of dezelfde) variabele. In het referentie handboek vindt u meer informatie over deze functie, en over het programmeren in machinetaal.

String functies

ASC(X$) (b.v. X = ASC(X$) geeft de ASCII code van het eerste karakter van X$.

CHR$(X) (b.v. X$ = CHR$(X))

Dit is juist het tegenovergestelde van ASC(X$). De functie geeft 1 karakter, datgene waarvan de ASCII waarde X is.

LEFT$(X$,X)

Geeft een string waarin de X meest linkse karakters van string X$ zitten.

LEN(X$)

Geeft het aantal karakters (inclusief spaties) in string X$.

MID$(X$,S,X)

Geeft een string bestaande uit X karakters, beginnend vanaf karakter no. S van string X$.

RIGHT$(X$,X)

Geeft een string waarin de X meest rechtse karakters van X$ zitten.

STR$(X)

Geeft een string gelijk aan hetgeen afgedrukt wordt als we het getal X afdrukken.

VAL(X$)

Deze functie converteert X$ in een getal. Het is eigenlijk het omgekeerde van STR$. De string wordt bekeken van links naar rechts, net zoveel karakters als bij een getal zouden kunnen horen, b.v.:

```
10 X = VAL("123.456")          X = 123.456
10 X = VAL("12A13B")           X = 12
10 X = VAL("R14017")           X = 0
10 X = VAL("-1.23.45.67")      X = 1.23
```

Overige functies

FRE(X) (b.v. A=FRE(X))

Geeft het aantal "vrije" geheugenplaatsen. De waarde van X is niet belangrijk in de CBM 64.

POS(X) (b.v. A=POS(X))

Geeft de kolompositie waar een volgende PRINT instructie zal gaan afdrukken (0....39) X kan ook hier elke waarde hebben.

SPC(X)

Wordt gebruikt in de PRINT instructie. We slaan dan X spaties over.

TAB(X)

TAB wordt ook alleen in PRINT instructies gebruikt. De te printen tekst verschijnt in kolom (X) van het scherm. TAB werkt niet met een printer.

AANHANGSEL D

AFKORTINGEN VAN BASIC WOORDEN

Om tijd te besparen bij het intikken van programma's en commando's kan men de meeste woorden van Basic afkorten. Zo is de afkorting voor PRINT een vraagteken: ?. De afkortingen voor de andere woorden bestaan uit de eerste of de eerste 2 letters van het woord gevolgd door de 3e letter met SHIFT ingedrukt. Tijdens het LISTen van het programma, verschijnen de woorden wel weer volledig. Sommige afkortingen bevatten ook het haakje links!

commando	afkorting	geeft op het scherm	commando	afkorting	geeft op het scherm
ABS	A SHIFT B	A ▕	END	E SHIFT N	E ╱
AND	A SHIFT N	A ╱	EXP	E SHIFT X	E ♣
ASC	A SHIFT S	A ♥	FN	NONE	FN
ATN	A SHIFT T	A ▏	FOR	F SHIFT O	F ▔
CHR$	C SHIFT H	C ▕	FRE	F SHIFT R	F ▁
CLOSE	CL SHIFT O	CL ▔	GET	G SHIFT E	G ▔
CLR	C SHIFT L	C ▁	GET#	NONE	GET#
CMD	C SHIFT M	C ╲	GOSUB	GO SHIFT S	GO ♥
CONT	C SHIFT O	C ▔	GOTO	G SHIFT O	G ▔
COS	NONE	COS	IF	NONE	IF
DATA	D SHIFT A	D ♠	INPUT	NONE	INPUT
DEF	D SHIFT E	D ▔	INPUT#	I SHIFT N	I ╱
DIM	D SHIFT I	D ╮	INT	NONE	INT

commando	afkorting	geeft op het scherm
LEFT$	LE SHIFT F	LE
LEN	NONE	LEN
LET	L SHIFT E	L
LIST	L SHIFT I	L
LOAD	L SHIFT O	L
LOG	NONE	LOG
MID$	M SHIFT I	M
NEW	NONE	NEW
NEXT	N SHIFT E	N
NOT	N SHIFT O	N
ON	NONE	ON
OPEN	O SHIFT P	O
OR	NONE	OR
PEEK	P SHIFT E	P
POKE	P SHIFT O	P
POS	NONE	POS
PRINT	?	?
PRINT#	P SHIFT R	P
READ	R SHIFT E	R
REM	NONE	REM
RESTORE	RE SHIFT S	RE ♥
RETURN	RE SHIFT T	RE
RIGHT$	R SHIFT I	R
RND	R SHIFT N	R
RUN	R SHIFT U	R
SAVE	S SHIFT A	S ♠
SGN	S SHIFT G	S
SIN	S SHIFT I	S
SPC(	S SHIFT P	S
SQR	S SHIFT Q	S ●
STATUS	ST	ST
STEP	ST SHIFT E	ST
STOP	S SHIFT T	S
STR$	ST SHIFT R	ST
SYS	S SHIFT Y	S
TAB(	T SHIFT A	T ♠
TAN	NONE	TAN
THEN	T SHIFT H	T
TIME	TI	TI
TIME$	TI$	TI$
USR	U SHIFT S	U ♥
VAL	V SHIFT A	V ♠
VERIFY	V SHIFT E	V
WAIT	W SHIFT A	W ♠

AANHANGSEL E

SCHERMCODES

In de volgende lijst vindt u alle ingebouwde karakters van de Commodore 64 karakterset. Achter de karakters staan de waarden welke in het schermgeheugen (lokaties 1024 - 2123) gePOKEd moeten worden om het gewenste karakter te krijgen. Omgekeerd natuurlijk ziet u welk teken overeenkomt met een waarde welke via PEEK is verkregen.

Er zijn twee verschilende karaktersets, maar er is er steeds maar een beschikbaar. Dit betekent dat het niet mogelijk is om karakters van set 1 tegelijkertijd met die van set 2 weer te geven. We schakelen van de ene naar de andere set over door het indrukken van de Commodore toets en SHIFT.

Vanuit een programma schakelen we over door POKE 53272,29 en POKE 53272,31. De tekens kunnen ook in diapositief worden weergegeven. De waarde van zo'n karakter (zg RVS karakter, of reverse) verkrijgen we door bij de genoemde waarden 128 op te tellen.

VOORBEELD

Indien u een balletje op lokatie 1504 weer wilt geven, POKE dan de waarde van een balletje (81) in lokatie 1504: POKE 1504,81.
Er is ook een geheugengebied waar de kleuren voor elke schermlokatie worden bepaald. (55296 - 56295). Om de kleur van de bal in geel te veranderen (kleurcode 7) POKEd u de corresponderende kleur lokatie (hier: 55776) met de kleur: POKE 55776,7.
Kijk in AANHANGSEL G voor de complete layout van scherm- en kleurgeheugen. Ook vindt u daar de codes voor de kleuren.

SCHERMCODES

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
@		0	C	c	3	F	f	6
A	a	1	D	d	4	G	g	7
B	b	2	E	e	5	H	h	8

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
I	i	9	%		37	♠	A	65
J	j	10	&		38	[graphic]	B	66
K	k	11	'		39	[graphic]	C	67
L	l	12	(		40	[graphic]	D	68
M	m	13	)		41	[graphic]	E	69
N	n	14	*		42	[graphic]	F	70
O	o	15	+		43	[graphic]	G	71
P	p	16	,		44	[graphic]	H	72
Q	q	17	–		45	[graphic]	I	73
R	r	18	.		46	[graphic]	J	74
S	s	19	/		47	[graphic]	K	75
T	t	20	0		48	[graphic]	L	76
U	u	21	1		49	[graphic]	M	77
V	v	22	2		50	[graphic]	N	78
W	w	23	3		51	[graphic]	O	79
X	x	24	4		52	[graphic]	P	80
Y	y	25	5		53	●	Q	81
Z	z	26	6		54	[graphic]	R	82
[		27	7		55	♥	S	83
£		28	8		56	[graphic]	T	84
]		29	9		57	[graphic]	U	85
↑		30	:		58	⊠	V	86
←		31	;		59	○	W	87
SPACE		32	<		60	♣	X	88
!		33	=		61	[graphic]	Y	89
"		34	>		62	♦	Z	90
#		35	?		63	⊞		91
$		36	[graphic]		64	[graphic]		92

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
		93			105			117
		94			106			118
		95			107			119
SPACE		96			108			120
		97			109			121
		98			110		✓	122
		99			111			123
		100			112			124
		101			113			125
		102			114			126
		103			115			127
		104			116			

De codes 128-255 zijn de codes 0-127 in diapositief.

AANHANGSEL F

ASCII EN CHR$ CODES

De volgende tabellen laten u zien welke karakters op het scherm verschijnen bij gebruik van de CHR$ functie. Omgekeerd vindt u de waarden welke we krijgen met de ASC functie als we intikken PRINT ASC("X") waarbij X elk in te tikken karakter kan zijn.
De lijst is van nut bij het verwerken van karakters welke via GET binnenkomen, bij het omzetten van grote naar kleine letters, of het afdrukken van karakters welke normaal niet tussen aanhaaltekens voor kunnen komen (b.v. het aanhaalteken " zelf!)

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
	0	CRSR ↓	17	"	34	3	51
	1	RVS ON	18	#	35	4	52
	2	CLR HOME	19	$	36	5	53
	3	INST DEL	20	%	37	6	54
	4		21	&	38	7	55
WHT	5		22	'	39	8	56
	6		23	(	40	9	57
	7		24	)	41	:	58
DISABLES SHIFT C=	8		25	*	42	;	59
ENABLES SHIFT C=	9		26	+	43	<	60
	10		27	,	44	=	61
	11	RED	28	−	45	>	62
	12	CRSR →	29	.	46	?	63
RETURN	13	GRN	30	/	47	@	64
SWITCH TO LOWER CASE	14	BLU	31	0	48	A	65
	15	SPACE	32	1	49	B	66
	16	!	33	2	50	C	67

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
D	68	♠	97	π	126	Grey 3	155
E	69		98		127	PUR	156
F	70		99		128	CRSR	157
G	71		100	Orange	129	YEL	158
H	72		101		130	CYN	159
I	73		102		131	SPACE	160
J	74		103		132		161
K	75		104	f1	133		162
L	76		105	f3	134		163
M	77		106	f5	135		164
N	78		107	f7	136		165
O	79		108	f2	137		166
P	80		109	f4	138		167
Q	81		110	f6	139		168
R	82		111	f8	140		169
S	83		112	SHIFT RETURN	141		170
T	84	●	113	SWITCH TO UPPER CASE	142		171
U	85		114		143		172
V	86	♥	115	BLK	144		173
W	87		116	CRSR	145		174
X	88		117	RVS OFF	146		175
Y	89	╳	118	CLR HOME	147		176
Z	90	○	119	INST DEL	148		177
[	91	♣	120	Brown	149		178
£	92		121	Lt. Red	150		179
]	93	♦	122	Grey 1	151		180
↑	94	┼	123	Grey 2	152		181
←	95		124	Lt. Green	153		182
	96		125	Lt. Blue	154		183

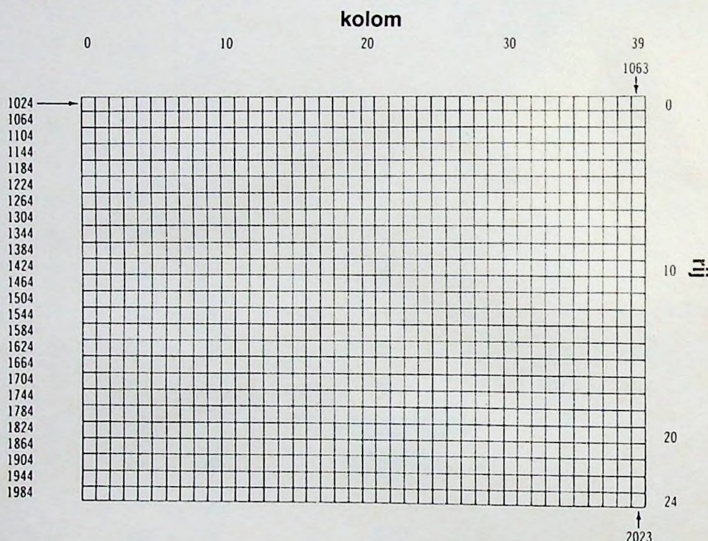
PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
	184		186		188		190
	185		187		189		191

De codes 192-223 zijn gelijk aan 96-127
De codes 224-254 zijn gelijk aan 160-190
De code 255 is hetzelfde als 126

AANHANGSEL G

DE LAYOUT VAN SCHERM- EN KLEUR GEHEUGEN

De volgende afbeeldingen geven aan welke geheugenplaatsen bepalend zijn voor de plaats van de karakters op het scherm, en welke kleur deze karakters zullen hebben.

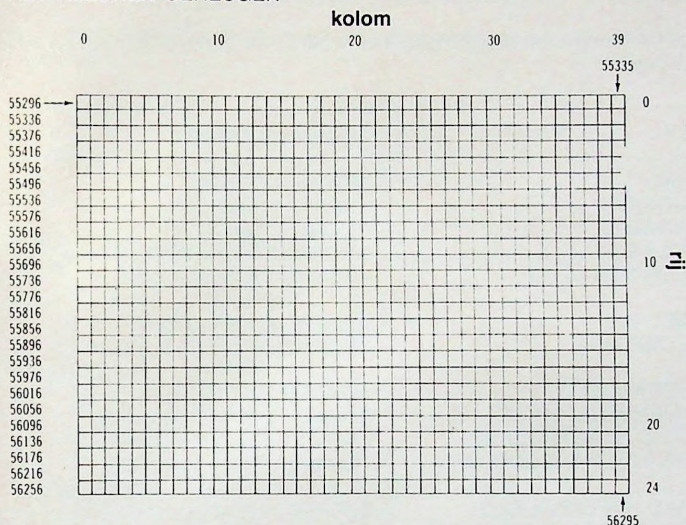


De waarden welke gePOKEd moeten worden in een geheugenplaats voor kleur zijn:

0 zwart	8 oranje
1 wit	9 bruin
2 rood	10 licht rood
3 cyaan	11 grijs 1
4 paars	12 grijs 2
5 groen	13 licht groen
6 blauw	14 licht blauw
7 geel	15 grijs 2

Om bijvoorbeeld de kleur van het karakter in de linkerbovenhoek in rood te veranderen, tikt u in: POKE 55296,2

HET KLEUREN GEHEUGEN



AANHANGSEL H

AFGELEIDE WISKUNDIGE FUNCTIES

Functies welke niet standaard voorkomen in uw Commodore 64 kunnen als volgt worden afgeleid:

Functie	**Equivalent in Basic**
SECANS	SEC(X)=1/COS(X)
COSECANS	CSC(X)=1/SIN(X)
COTANGENS	COT(X)=1/TAN(X)
ARCSINUS	ARCSIN(X)=ATN(X/SQR(-X*X+1))
ARCCOSINUS	ARCCOS(X)=-ATN(X/SQR (-X*X +1))+pi/2
ARCSECANS	ARCSEC(X)=ATN(X/SQR(X*X-1))
ARCCOSECANS	ARCCSC(X)=ATN(X/SQR(X*X-1)) +(SGN(X)-pi/2
ARCCOTANGENS	ARCOT(X)=ATN(X)+pi/2
SINUS HYPERBOLICUS	SINH(X)=(EXP(X)-EXP(-X))/2
COSINUS HYPERBOLICUS	COSH(X)=(EXP(X)+EXP(-X))/2
TANGENS HYPERBOLICUS	TANH(X)=EXP(-X)/(EXP(X) +EXP(-X))*2+1
SECANS HYPERBOLICUS	SECH(X)=2/(EXP(X)+EXP(-X))
COSECANS HYPERBOLICUS	CSCH(X)=2/(EXP(X)-EXP(-X))
COTANGENS HYPERBOLICUS	COTH(X)=EXP(-X)/(EXP(X) -EXP(-X)*2+1
ARCSINUS HYPERBOLICUS	ARCSINH(X)=LOG(X+SQR(X*X+1))
ARCCOSINUS HYPERBOLICUS	ARCCOSH(X)=LOG(X+SQR(X*X-1))
ARCTANGENS HYPERBOLICUS	ARCTANH(X)=LOG((1+X)/(1-X))/2
ARCSECANS HYPERBOLICUS	ARCSECH(X)=LOG((SQR (-X*X+1)+1/X)
ARCCOSECANS HYPERBOLICUS	ARCCSCH(X)=LOG((SGN(X)*SQR (X*X+1/X)
ARCCOTANGENS HYPERBOLICUS	ARCCOTH(X)=LOG((X+1)/(X-1))/2

AANHANGSEL I

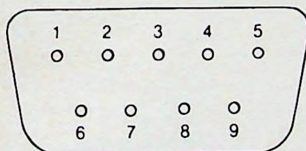
AANSLUITGEGEVENS

In dit AANHANGSEL vindt u een opsomming van de aansluitgegevens van de connectors.

1. de spelconnectors
2. de cartridge connector
3. audio-video
4. de seriële bus
5. de hoogfrequent uitgang
6. cassette poort
7. user poort

Control port 1 (Spel connector 1)

Pin	Type	NOOT
1	JOYAO	
2	JOYA1	
3	JOYA2	
4	JOYA3	
5	POT AY	
6	BUTTON A/LP	
7	+5V	MAX. 50mA
8	GND	
9	POT AX	



Control port 2 (Spel connector 2)

Pin	Type	NOOT
1	JOYB0	
2	JOYB1	
3	JOYB2	
4	JOYB3	
5	POT BY	
6	BUTTON B	
7	+5V	MAX. 50mA
8	GND	
9	POT BX	

Cartridge Expansion Slot

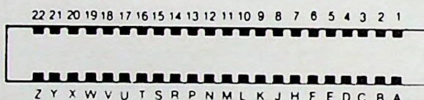
Pin	Type
1	GND
2	+ 5V
3	+ 5V
4	IRQ
5	R/W
6	Dot Clock
7	I/O 1
8	GAME
9	EXROM
10	I/O 2
11	ROML

Pin	Type
A	GND
B	ROMH
C	RESET
D	NMI
E	S 02
F	A15
H	A14
J	A13
K	A12
L	A11
M	A10

Cartridge Expansion Slot

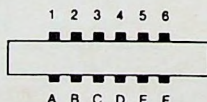
Pin	Type
12	BA
13	DMA
14	D7
15	D6
16	D5
17	D4
18	D3
19	D2
20	D1
21	D0
22	GND

Pin	Type
N	A9
P	A8
R	A7
S	A6
T	A5
U	A4
V	A3
W	A2
X	A1
Y	A0
Z	GND



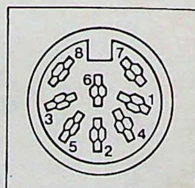
Cassette

Pin	Type
A-1	GND
B-2	+5V
C-3	CASSETTE MOTOR
D-4	CASSETTE READ
E-5	CASSETTE WRITE
F-6	CASSETTE SENSE



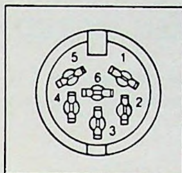
Audio/Video: 8 Pin

Pin	Type	Noot
1	LUM/SYNC	Luminantie/SYNC uitgang (zwart/wit)
2	GND	Afscherming/massa
3	AUDIO OUT	Geluid uit
4	VIDEO OUT	Composiet signaal uitgang
5	AUDIO IN	Geluid in
6	COLOR OUT	Chroma signaal uit (kleur)
7	NC	Niet aangesloten
8	NC	Niet aangesloten



Serieele bus (I/O)

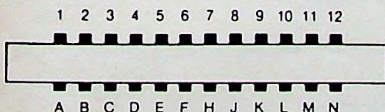
Pin	Type
1	SERIAL SRQIN
2	GND
3	SERIAL ATN IN/OUT
4	SERIAL CLK IN/OUT
5	SERIAL DATA IN/OUT
6	RESET



User I/O

Pin	Type	Note
1	GND	
2	+ 5V	MAX. 100mA
3	RESET	
4	CNTI	
5	SP1	
6	CNT2	
7	SP2	
8	PC2	
9	SER. ATN IN	
10	9 VAC	MAX. 100mA
11	9 VAC	MAX. 100mA
12	GND	

Pin	Type	Note
A	GND	
B	FLAG2	
C	PB0	
D	PB1	
E	PB2	
F	PB3	
H	PB4	
J	PB5	
K	PB6	
L	PB7	
M	PA2	
N	GND	



AANHANGSEL J

HET PROBEREN WAARD

De programma's op de pagina's 148 tot 150 zijn aardig om een diepergaande kennis over uw Commodore 64 op te doen.

listing jotto

```
100 PRINT"█UOTTO                  JIM BUTTERFIELD"
120 INPUT"█WANT INSTRUCTIONS";Z$:IFASC(Z$)=78GOTO250
130 PRINT"█TRY TO GUESS THE MYSTERY 5-LETTER WORD"
140 PRINT"█YOU MUST GUESS ONLY LEGAL 5-LETTER"
150 PRINT"█WORDS, TOO..."
160 PRINT"█YOU WILL BE TOLD THE NUMBER OF MATCHES"
170 PRINT"(OR 'JOTS') OF YOUR GUESS."
180 PRINT"█HINT: THE TRICK IS TO VARY SLIGHTLY"
190 PRINT"       FROM ONE GUESS TO THE NEXT; SO THAT"
200 PRINT"       IF YOU GUESS 'BATCH' AND GET 2 JOTS"
210 PRINT"       YOU MIGHT TRY 'BOTCH' OR 'CHART'"
220 PRINT"       FOR THE NEXT GUESS..."
250 DATA BXBSF,IPCCZ,DBDIF,ESFBE,PGGBM
260 DATA HPSHF,IBUDI,DJWJM,KPMMZ,IBZBL
270 DATA SBKBI,MFWFM,NJNJD,BOOFY,QJQFS
280 DATA RVFTU,SJWFS,QSFTT,PUUFS,FWFOU
290 DATA XFBWF,FYUPM,NVTIZ,AFCSB,GJAAZ
300 DATA UIJDL,ESVOL,GMPPE,UJHFS,GBLFS
310 DATA CPPUI,MZJOH,TRVBU,HBVAF,PXJOH
320 DATA UISFF,TJHIU,BYMFT,HSVNQ,BSFAB
330 DATA RVBSU,DSFFQ,CFMDI,QSFTT,TQBSL
340 DATA SBEBS,SVSBM,TNFMM,GSPXO,ESJGU
400 N=50
410 DIM N$(N),Z(5),Y(5)
420 FORJ=1TON:READN$(J):NEXTJ
430 T=TI
440 T=T/1000:IFT>=1THENGOTO440
450 Z=RND(-T)
500 G=0:N$=N$(RND(1)*N+1)
510 PRINT"I HAVE A FIVE LETTER WORD:":IFR>0GOTO560
520 PRINT"GUESS (WITH LEGAL WORDS)"
530 PRINT"AND I'LL TELL YOU HOW MANY"
540 PRINT"'JOTS', OR MATCHING LETTERS,"
550 PRINT"YOU HAVE...."
560 G=G+1:INPUT"YOUR WORD ";Z$
570 IF LEN(Z$)<>5THENPRINT"YOU MUST GUESS A 5-LETTER WORD!":GOTO560
580 V=0:H=0:M=0
590 FORJ=1TO5
600 Z=ASC(MID$(Z$,J,1)):Y=ASC(MID$(N$,J,1))-1:IFY=64THENY=90
610 IFZ<65ORZ>90THENPRINT"DIT IS GEEN WOORD!":GOTO560
620 IFZ=65ORZ=69ORZ=73ORZ=79ORZ=85ORZ=89THENV=V+1
630 IFZ=YTHENM=M+1
640 Z(J)=Z:Y(J)=Y:NEXTJ
650 IFM=5GOTO800
660 IFV=0ORV=5THENPRINT"COME ON..WHAT KIND OF A WORD IS THAT?":GOTO560
670 FORJ=1TO5:Y=Y(J)
680 FORK=1TO5:IFY=Z(K)THENH=H+1:Z(K)=0:GOTO700
690 NEXTK
700 NEXTJ
710 PRINT"██████████████████████";K;"JOTS" :FORII=1TO5:PRINTCHR$(Y(II));:NEXT:PRI
NT
720 IFG<30GOTO560
730 PRINT"I'D BETTER TELL YOU.. THE WORD WAS '"
```

```
740 FORJ=1TO5:PRINTCHR$(Y(J));:NEXTJ
750 PRINT"'":GOTO810
800 PRINT"YOU GOT IT IN ONLY";G;"GUESSES"
810 INPUT"▒ANOTHER WORD";Z$
820 R=1:IFASC(Z$)<>78GOTO500
READY.
```

listing hutspot

```
1 REM SEQUENCE
2 REM
3 REM *** VAN DE PET USER GROUP
4 REM *** SOFTWARE EXCHANGE
5 REM *** POSTBUS 371
6 REM *** MONTGOMMERYVILLE, PA 18936 USA.
7 REM
8 REM
50 DIM A$(26)
100 Z$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
110 Z1$="12345678901234567890123456"
200 PRINT"▒▒▒LENGTE VAN DE STRING DIE AFGEZOCHT MOET WORDEN▒"
220 INPUT"MAXIMUM LENGTE IS 26 ";S%
230 IFS%<1ORS%>26THEN200
240 S=S%
300 FORI=1TOS
310 A$(I)=MID$(Z$,I,1)
320 NEXTI
400 REM RANDOMIZE STRING
420 FORI=1TOS
430 K=INT(RND(1)*S+1)
440 T$=A$(I)
450 A$(I)=A$(K)
460 A$(K)=T$
470 NEXTI
480 GOSUB950
595 T=0
600 REM KEER SUBSTRING OM
605 T=T+1:PRINTT
610 INPUT"HOEVEEL KEER OMDRAAIEN";R%
620 IFR%=0THEN 900
630 IFR%>0AND R%<=S THEN650
640 PRINT"MOET TUSSEN 1 EN";S;"ZIJN":GOTO 610
650 R=INT(R%/2)
660 FORI=1TOR
670 T$=A$(I)
680 A$(I)=A$(R%-I+1)
690 A$(R%-I+1)=T$
700 NEXTI
750 GOSUB950
800 C=1:FORI=2TOS
810 IFA$(I)=>A$(I-1)THEN830
820 C=0
830 NEXTI
840 IFC=0THEN600
850 PRINT"▒JE HEBT HET IN";T;"POGING(EN) GEDAAN"
900 REM TEST VOOR VOLGEND SPEL
910 INPUT"▒WIL JE EEN ANDER SPEL ";Y$
920 IFLEFT$(Y$,1)="J" OR Y$="OK" OR Y$="1" THEN200
930 END
950 PRINT
960 PRINTLEFT$(Z1$,S)
970 FORI=1TOS:PRINTA$(I);:NEXT I
980 PRINT"▒"
990 RETURN
READY.
```

Listing piano

```
10 REM PIANO                        080384 EMJ
20 :
100 PRINT"■ ■ ■ ■ | ■ ■ ■ | ■ ■ | ■ ■ "
110 PRINT" ■ ■ ■ | ■ ■ ■ | ■ ■ | ■ ■ "
120 PRINT" ■ ■ ■ | ■ ■ ■ | ■ ■ | ■ ■ "
130 PRINT" ■ | | | | | | | | | | | | "
140 PRINT" ■Q |W |E |R |T |Y |U |I |O |P |@ |* |↑"
150 PRINT"■'SPATIE' VOOR PEDAAL"
160 PRINT"■'F1,F3,F5,F7' OCTAAF SELECTIE"
170 PRINT"■'F2,F4,F6,F8' GOLFVORM■"
180 PRINT"TOETS '?' VOOR EINDE                    EEN OGENBLIKJE..."
190 S=13*4096+1024:DIMF(26):DIMK(255)
200 FORI=0TO28:POKES+I,0:NEXT
210 F1=7040:FORI=1TO26:F(27-I)=F1*5.8+30:F1=F1/2↑(1/12):NEXT
220 K$="Q2W3ER5T6Y7UI9O0P@-*£↑"
230 FORI=1TOLEN(K$):K(ASC(MID$(K$,I)))=I:NEXT
240 PRINT"■                                        "
250 AT=0:DE=0:SU=15:RE=9:SV=SU*16+RE:AV=AT*16+DE:WV=16:W=0:M=1:OC=4:HB=256:Z=0
260 FORI=0TO2:POKES+5+I*7,AT*16+DE:POKES+6+I*7,SU*16+RE
270 POKES+2+I*7,4000AND255:POKES+3+I*7,4000/256:NEXT
280 POKES+24,15:REM+16+64:POKES+23,7
300 GETA$:IFA$=""THEN300
305 IF A$="?" THEN GOTO 1000
310 FR=F(K(ASC(A$)))/M:T=V*7:CR=S+T+4:IFFR=ZTHEN500
320 POKES+6+T,Z:REM STEL DEC/SUS IN
325 POKES+5+T,Z:REM STEL ATT/REL IN
330 POKECR,8:POKECR,0:REM TRIGGER
340 POKES+T,FR-HB*INT(FR/HB):REM STEL LO FREQ IN
350 POKES+1+T,FR/HB:REM STEL HI FREQ IN
360 POKES+6+T,SV:REM STEL DEC/SUS IN
365 POKES+5+T,AV:REM STEL ATT/REL IN
370 POKECR,WV+1:FORI=1TO50*AT:NEXT
375 POKECR,WV:REM PULS
380 IFP=1THENV=V+1:IFV=3THENV=0
400 GOTO300
500 IFA$="■"THENM=1:OC=4:GOTO300
510 IFA$="■"THENM=2:OC=3:GOTO300
520 IFA$="■"THENM=4:OC=2:GOTO300
530 IFA$="■"THENM=8:OC=1:GOTO300
540 IFA$="■"THENW=0:WV=16:GOTO300
550 IFA$="■"THENW=1:WV=32:GOTO300
560 IFA$="■"THENW=2:WV=64:GOTO300
570 IFA$="■"THENW=3:WV=128:GOTO300
580 IFA$=" "THENP=1-P:GOTO300
590 IFA$="■"THEN200
600 GOTO300
1000 S=13*4096+1024
1010 FORI=0TO28:POKES+I,0:NEXT:CLR
1020 END
READY.
```

AANHANGSEL K

HET OMZETTEN VAN PROGRAMMA'S NAAR COMMODORE BASIC

Indien u beschikt over programmatuur welke in een ander BASIC dialect is geschreven, dan zal het meestal niet moeilijk zijn om die programma's in Commodore BASIC om te zetten. Hier volgen aanwijzingen:

String afmetingen

Het is bij Commodore Basic niet nodig de lengte van een string van te voren op te geven. Verwijder daarom alle instructies die nodig waren om die lengte te specificeren. Een instructie als b.v. DIM A$(I,J) welke gebruikt wordt om J elementen van lengte I te dimensioneren in een matrix moet omgezet worden in de Commodore Basic instructie DIM A$(J). Sommige basics gebruiken een komma of het & teken om strings samen te voegen. Commodore Basic gebruikt hiervoor het plus (+) teken. Verder gebruikt Commodore Basic de MID$, LEFT$ en RIGHT$ functies voor het maken van substrings. Basics welke bv A$(1) zetten om het 1e karakter van string A$ om te zetten of A$(I,J) om een string van J karakters te vormen vanaf karakter I in string A$ dienen dus als volgt aangepast te worden:

BASIC b.v.	Commodore 64 Basic
A$(I)= X$	A$=LEFT$(A$,I-1)+X$+RIGHT$(A$,LEN(A$)-1)
A$(I,J)=X$	A$=LEFT$(A$,I-1)+X$+RIGHT$(A$,LEN(A$)-1-J)

Meervoudige waardetoekenning

Sommige Basics laten instructie toe als

```
10 LET B=C=0
```

om meer variabelen tegelijkertijd een waarde toe te kennen. In dit geval zou Commodore Basic het 2e "is gelijk" teken als een logische operator beschouwen en aan B de waarde -1 toekennen als C = 0. We moeten een dergelijke uitdrukking dus wijzigen in:

```
10 C=0 : B=0
```

Meer instructies op een regel

Sommige Basics gebruiken een backslash (/) om meer instructies op een regel te scheiden. Verander die backslash in een dubbele punt (:)

Mat functies

Programma's welke de MAT functie gebruiken moeten worden gewijzigd. Gebruik een FOR....NEXT lus om de matrix te vullen.

AANHANGSEL L

FOUT BOODSCHAPPEN

In het geval u een fout gemaakt heeft, stelt uw Commodore 64 u daarvan op de hoogte. Hieronder vindt u een lijst van foutmeldingen, met de mogelijke oorzaak.

BAD DATA	Er kwam een string uit een bestand binnen, en het programma verwachtte numerieke data.
BAD SUBSCRIPT	Het programma probeerde een matrix element te bewerken dat niet was gedimensioneerd.
CAN'T CONTINUE	In dit geval werkt de CONT instructie niet omdat het programma nooit geRUNd had, omdat er een fout was, of omdat er ondertussen iets aan het programma werd veranderd.
DEVICE NOT PRESENT	Spreekt voor zichzelf. Deze foutmelding komt als men via OPEN, CLOSE, CMD, PRINT#, INPUT# of GET# een toestel benadert dat niet beschikbaar is.
DIVISION BY ZERO	Oftewel delen door nul, is niet toegestaan.
EXTRA IGNORED	Komt als u meer antwoorden geeft dan de computer aan u vraagt. Komma's in antwoorden kunnen dergelijke boodschappen veroorzaken!
FILE NOT FOUND	Krijgt u als u naar een bestand of programma op tape zoekt dat er niet was voor een merkteken "bandeinde". Vanaf disk krijgt u deze melding als het gewenste programma of bestand niet op de disk staat.
FILE NOT OPEN	Is de boodschap als u probeert via CMD, PRINT#, INPUT# OF GET# een bestand te benaderen dat niet eerst geopend was.
FILE OPEN	Hier probeert u een reeds geopende file een 2e keer te openen.
FORMULA TOO COMPLEX	Probeert u de string formule die verwerkt wordt in 2 delen te splitsen.
ILLEGAL DIRECT	INPUT-instructies mogen alleen in een programma gebruikt worden!
ILLEGAL QUANTITY	De parameter van een functie is hier buiten het toegestane bereik, b.v.. SQR(-2) of LOG(-1).
LOAD	Hier is een probleem met de tape of cassette
NEXT WITHOUT FOR	Deze fout komt als een lus incorrect is, of als u verwijst naar een variabele achter NEXT welke niet gelijk is aan die waarmee u bij FOR bent gestart.
NOT INPUT FILE	Hier probeert de gebruiker INPUT of GET te gebruiken vanaf een file of bestand dat geopend was om te worden beschreven.

NOT OUTPUT FILE	De gebruiker probeerde te PRINTen naar een file of bestand dat geopend was om gelezen te worden.
OUT OF DATA	Deze foutboodschap verschijnt wanneer men met een READ instructie meer gegevens probeert te lezen dan er in DATA regels stonden. Deze fout zult u ook zien als de computer over READY stond en u drukt op RETURN: READ Y
OUT OF MEMORY	Verschijnt als er geen RAM geheugen meer beschikbaar is voor het programma of de variabelen ervan. Deze boodschap kan ook komen wanneer er te veel FOR....NEXT loops binnen een FOR NEXT loop zitten, of wanneer er meer dan 21 GOSUB's in een subroutine zitten.
OVERFLOW	Het grootste toegelaten getal is 1.70141884 E+38. Indien deze waarde wordt overschreden, verschijnt deze melding.
REDIM'D ARRAY	Een matrix of ARRAY mag maar eenmalig worden gedimensioneerd. De fout verschijnt echter meestal doordat matrix elementen gebruikt worden voor dimensionering plaatsvond. In zo'n geval wordt het aantal elementen op 11 gesteld. Bij een latere DIM krijgen we dan de melding.
REDO FROM START	Komt als er karakters ingetikt worden in plaats van cijfers als er een numerieke waarde ingegeven moet worden. Dit is geen fatale fout. Tik de juiste gegevens in: het programma zal normaal verderlopen.
RETURN WITHOUT GOSUB	Het programma kwam RETURN tegen zonder dat het daar kwam via een GOSUB instructie.
STRING TOO LONG	Een string kan maximaal 255 karakters bevatten.
? SYNTAX ERROR	Waarschijnlijk de fout die u met OUT OF DATA het meest zult zien. Er is een spelfout, een haakje dat ontbreekt, een haakje te veel ......
TYPE MISMATCH	Deze fout treedt op als een getal gebruikt wordt in plaats van een string, of omgekeerd.
UNDEF'D FUNCTION	Er werd gebruik gemaakt van een functie welke nog niet via DEF FN was gedefinieerd.
UNDEF'D STATEMENT	Treedt op als men met ON, GOTO of GOSUB wil springen naar een niet bestaande regel.
VERIFY	Het programma op tape of disk is niet identiek aan dat in het geheugen.

AANHANGSEL M

MUZIEKNOTEN

Hieronder vindt u een complete lijst van de waarden welke in de toonhoogte registers gePOKEd moeten worden voor het produceren van de muziek noten tussen C-0 en A#-7.

Pokewaarden van noten

MUZIEKNOOT		OSCILLATOR FREQ		
NOOT	OCTAAF	DECIMAAL	HI-BYTE	LO-BYTE
0	C–0	268	1	12
1	C#–0	284	1	28
2	D–0	301	1	45
3	D#–0	318	1	62
4	E–0	337	1	81
5	F–0	358	1	102
6	F#–0	379	1	123
7	G–0	401	1	145
8	G#–0	425	1	169
9	A–0	451	1	195
10	A#–0	477	1	221
11	B–0	506	1	250
16	C–1	536	2	24
17	C#–1	568	2	56
18	D–1	602	2	90
19	D#–1	637	2	125
20	E–1	675	2	163
21	F–1	716	2	204
22	F#–1	758	2	246
23	G–1	803	3	35
24	G#–1	851	3	83
25	A–1	902	3	134
26	A#–1	955	3	187
27	B–1	1012	3	244
32	C–2	1072	4	48

MUZIEKNOOT		OSCILLATOR FREQ		
NOOT	OCTAAF	DECIMAAL	HI-BYTE	LO-BYTE
33	C#−2	1136	4	112
34	D−2	1204	4	180
35	D#−2	1275	4	251
36	E−2	1351	5	71
37	F−2	1432	5	152
38	F#−2	1517	5	237
39	G−2	1607	6	71
40	G#−2	1703	6	167
41	A−2	1804	7	12
42	A#−2	1911	7	119
43	B−2	2025	7	233
48	C−3	2145	8	97
49	C#−3	2273	8	225
50	D−3	2408	9	104
51	D#−3	2551	9	247
52	E−3	2703	10	143
53	F−3	2864	11	48
54	F#−3	3034	11	218
55	G−3	3215	12	143
56	G#−3	3406	13	78
57	A−3	3608	14	24
58	A#−3	3823	14	239
59	B−3	4050	15	210
64	C−4	4291	16	195
65	C#−4	4547	17	195
66	D−4	4817	18	209
67	D#−4	5103	19	239
68	E−4	5407	21	31
69	F−4	5728	22	96
70	F#−4	6069	23	181
71	G−4	6430	25	30
72	G#−4	6812	26	156
73	A−4	7217	28	49
74	A#−4	7647	29	223
75	B−4	8101	31	165
80	C−5	8583	33	135
81	C#−5	9094	35	134

MUZIEKNOOT		OSCILLATOR FREQ		
NOOT	OCTAAF	DECIMAAL	HI-BYTE	LO-BYTE
82	D–5	9634	37	162
83	D#–5	10207	39	223
84	E–5	10814	42	62
85	F–5	11457	44	193
86	F#–5	12139	47	107
87	G–5	12860	50	60
88	G#–5	13625	53	57
89	A–5	14435	56	99
90	A#–5	15294	59	190
91	B–5	16203	63	75
96	C–6	17167	67	15
97	C#–6	18188	71	12
98	D–6	19269	75	69
99	D#–6	20415	79	191
100	E–6	21629	84	125
101	F–6	22915	89	131
102	F#–6	24278	94	214
103	G–6	25721	100	121
104	G#–6	27251	106	115
105	A–6	28871	112	199
106	A#–6	30588	119	124
107	B–6	32407	126	151
112	C–7	34334	134	30
113	C#–7	36376	142	24
114	D–7	38539	150	139
115	D#–7	40830	159	126
116	E–7	43258	168	250
117	F–7	45830	179	6
118	F#–7	48556	189	172
119	G–7	51443	200	243
120	G#–7	54502	212	230
121	A–7	57743	225	143
122	A#–7	61176	238	248
123	B–7	64814	253	46

FILTER SETTINGS

Location	Contents
54293	Low cutoff frequency (0–7)
54294	High cutoff frequency (0–255)
54295	Resonance (bits 4–7) Filter voice 3 (bit 2) Filter voice 2 (bit 1) Filter voice 1 (bit 0)
54296	High pass (bit 6) Bandpass (bit 5) Low pass (bit 4) Volume (bits 0–3)

AANHANGSEL N

Er zijn een aantal tijdschriften over de Commodore computer welke steeds de laatste nieuwtjes voor uw Commodore 64 beschrijven. De 4 meest bekende zijn:

Commodore

Verschijnt maandelijks. Het is een uitgave van Commodore U.S.A.

VCGN nieuws

Uitgave van Vereniging van Commodore gebruikers Nederland (VCGN) secretariaat De Brink 928, 2553 HT Den Haag

HCC nieuwsbrief

Uitgave van HCC Nederland, Postbus 149, 2250 AC Voorschoten. HCC is een algemeen georienteerde computer club.

PBE-blad

Verschijnt 8 x per jaar en is een uitgave van Copytronix, Burgemeester van Suchtelenstraat 46, 7413 XP te Deventer.

POWER PLAY

Verschijnt 4 x per jaar.

Literatuuropgave

Boeken over de Commodore 64.

Commodore 64 leren programmeren.

ISBN 90 6082 252 8. De Muiderkring B.V. M.B. Immerzeel.

Nederlandstalig boek, gaat in op het programmeren in BASIC op de Commodore 64. Bevat naast een beschrijving van de computer zelf ook een groot aantal voorbeeld programma's.

Graphic Art On The Commodore 64.

ISBN 0 946408 15 7. Sunshine Books. Boris Allen.

Engelstalig. Behandelt High Resolution Graphics (Turtle Graphics) Bevat veel voorbeelden in BASIC voor het creëren van High Resolution Graphics, ook in kleur.

Commodore Programmer's Reference Guide (for the CBM-64)

ISBN 0 672 22056 3. Commodore/Howard W. Sams $ Co. Inc.

Engelstalig. Alle gegevens over mogelijkheden, opbouw en programmerings instrukties; specifikaties van chips en complete schematuur.

64-Intern

Data Becker,Angerhausen,Becker,Englisch & Gerits

Duits. Aanbevolen naast de CBM-64 Reference manual. In dit boek zijn de listings van BASIC en KERNAL, inclusief kommentaar opgenomen. Biedt veel additionele informatie over de Commodore-64.

64 Tips & Tricks

Data Becker,Angerhausen,Englisch & Gerits

Duits. In dit boek zijn vele praktische voorbeelden opgenomen van toepassingen van de Commodore-64. Een apart hoofdstuk is gewijd aan CP/M op de 64. Maar ook het aansluiten van een Centronics printer met de daarbij behorende software wordt besproken.

The Complete Commodore 64 ROM Disassembly

ISBN 0-7156-1835-0.Peter Gerrard

Engelstalig. Memory maps, 6510 instruktieset en de komplete ROM listings met kommentaar.

The Working Commodore 64

ISBN 0 946408 02.5.David Lawrence

Engelstalig. Een samenvatting van allerlei subroutines en programma's .

Commodore 64 Machine Code Master

ISBN 0 946408 05 x.David Lawrence & Mark England

Engelstalig. Een samenvatting van machinetaal routines, o.a. om de instruktieset van de computer uit te breiden met nieuwe kommando's.

Commodore 64 Adventures.

ISBN 0 946408 11 4.Mike Grace

Engelstalig. Een gids voor het spelen en schrijven van avonturen spellen.

Toepassingen en spellen voor de Commodore 64

ISBN 90 201 1724 6.Max Voorburg

Een zeer gevariëerde verzameling BASIC programma's, zowel edukatief, zakelijk of algemeen.

Commodore 64

ISBN 90 201 1699 1.A. Sickler

Praktische tips, beschrijvingen, programma's etc

Beginners Assembly Language Programming for the CBM64

ISBN 0 90772 09 x.Derek Bush & Peter Holmes

Engelstalig. Instruktie en achtergronden van het werken met machinetaal. Bevat naast programma voorbeelden ook een cassette met daarop een Two-pass assembler.

Basic met de Commodore 64

ISBN 90-6215-092-6.L.R. Carter en E. Huzan

Ned. Uitleg van Basic, programmavoorbeelden, toepassing van kleur en geluid.

AANHANGSEL O

SPRITE REGISTERS

Register Dec	# Hex	D87	D86	D85	D84	D83	D82	D81	D80	
0	0	SOX7							S0XO	SPRITE O X Component
1	1	SOY7							S0YO	SPRITE O Y Component
2	2	S1X7							S1XO	SPRITE 1 X
3	3	S1Y7							S1YO	SPRITE 1 Y
4	4	S2X7							S2XO	SPRITE 2 X
5	5	S2Y7							S2YO	SPRITE 2 Y
6	6	S3X7							S3XO	SPRITE 3 X
7	7	S3Y7							S3YO	SPRITE 3 Y
8	8	S4X7							S4XO	SPRITE 4 X
9	9	S4Y7							S4YO	SPRITE 4 Y
10	A	S5X7							S5XO	SPRITE 5 X
11	B	S5Y7							S5YO	SPRITE 5 Y
12	C	S6X7							S6XO	SPRITE 6 X
13	D	S6Y7							S6YO	SPRITE 6 Y
14	E	S7X7							S7XO	SPRITE 7 X Component
15	F	S7Y7							S7YO	SPRITE 7 Y Component
16	10	S7X8	S6X8	S5X8	S4X8	S3X8	S2X8	S1X8	S0X8	MSB of X COORD.
17	11	RCB	ECM	BMM	BLNK	RSEL	YSCL2	YSCL1	YSCL0	Y SCROLL MODE
18	12	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	RASTER
19	13	LPX7							LPX0	LIGHT PEN X
20	14	LPY7							LPY0	LIGHT PEN Y

Sprite registers

Register Dec	# Hex	D87	D86	D85	D84	D83	D82	D81	D80	
21	15	SE7							SE0	SPRITE ENABLE (ON/OFF)
22	16	N.C.	N.C.	RST	MCM	CSEL	XSCL2	XSCL1	XSCL0	X SCROLL MODE
23	17	SEXY7							SEXY0	SPRITE EXPAND Y
24	18	VS13	VS12	VS11	VS10	CB13	CB12	CB11	N.C.	SCREEN Character Memory
25	19	IRQ	N.C.	N.C.	N.C.	LPIRQ	ISSC	ISBC	RIRQ	Interupt Request's
26	1A	N.C.	N.C.	N.C.	N.C.	MLPI	MISSC	MISBC	MRIRQ	Interupt Request MASKS
27	1B	BSP7							BSP0	Background-Sprite PRIORITY
28	1C	SCM7							SCMO	Multicolor SPRITE SELECT
29	1D	SEXX7							SEXXO	SPRITE EXPAND X
30	1E	SSC7							SSCO	Sprite-Sprite COLLISION
31	1F	SBC7							SBCO	Sprite-Background COLLISION

Register # Dec	Hex	Color
32	20	BORDER COLOR
33	21	BACKGROUND COLOR 0
34	22	BACKGROUND COLOR 1
35	23	BACKGROUND COLOR 2
36	24	BACKGROUND COLOR 3
37	25	SPRITE MULTICOLOR 0
38	26	SPRITE MULTICOLOR 1

Register # Dec	Hex	Color
39	27	SPRITE 0 COLOR
40	28	SPRITE 1 COLOR
41	29	SPRITE 2 COLOR
42	2A	SPRITE 3 COLOR
43	2B	SPRITE 4 COLOR
44	2C	SPRITE 5 COLOR
45	2D	SPRITE 6 COLOR
46	2E	SPRITE 7 COLOR

Dec	Hex	Color
0	0	BLACK
1	1	WHITE
2	2	RED
3	3	CYAN
4	4	PURPLE
5	5	GREEN
6	6	BLUE
7	7	YELLOW

Dec	Hex	Color
8	8	ORANGE
9	9	BROWN
10	A	LT. RED
11	B	GRAY 1
12	C	GRAY 2
13	D	LT. GREEN
14	E	LT. BLUE
15	F	GRAY 3

In multicolor mode kunnen alleen de kleuren 0...7 gebruikt worden.

AANHANGSEL P

MUZIEK REGISTERS VAN DE COMMODORE 64

De tabellen op de volgende bladzijden zijn erg handig voor het aflezen van de waarden welke u nodig heeft om een goed gebruik te maken van de 3 stemmen in uw Commodore 64. De procedure is simpel. In de eerste kolom vindt u steeds de actie, in de 2e kolom het POKE-adres en in de 3e en volgende kolommen de verschillende waarden.
B.v. POKE 54276,17. Kiest een driehoeksgolf voor stem 1.
Vergeet niet het volume in te stellen, anders kunt u geen geluid produceren. Het maximum volume voor alle 3 stemmen wordt ingesteld door middel van adres 54296, b.v. POKE 54296,15 geeft max. volume.
Voor het voortbrengen van een toon zijn er 2 instellingen nodig. POKE 54273,34 : POKE 54272,75 betekent toon C-5 voor stem 1.
Verder is uw keus niet beperkt tot de getallen van de tabellen. Als de toon "vals" klinkt, kunt u hem aanpassen: 35 in plaats van 34 bijvoorbeeld.
Hoger nagalmvolume of aanzweltijd verkrijgt u door waarden op te tellen. (voorbeeld: POKE 54277,96 is een combinatie van 32 en 64 voor de aanzwelsnelheid.......maar.....POKE 54277,20 geeft een lage aanzwelsnelheid (16) en een middelmatig verval (4).

	ADDRESS					REG #	DATA								REG NAME	REG
	A_4	A_3	A_2	A_1	A_0	(HEX)	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	**Voice 1**	TYPE
0	0	0	0	0	0	00	F_7	F_6	F_5	F_4	F_3	F_2	F_1	F_0	FREQ LO	WRITE-ONLY
1	0	0	0	0	1	01	F_{15}	F_{14}	F_{13}	F_{12}	F_{11}	F_{10}	F_9	F_8	FREQ HI	WRITE-ONLY
2	0	0	0	1	0	02	PW_7	PW_8	PW_5	PW_4	PW_3	PW_2	PW_1	PW_0	PW LO	WRITE-ONLY
3	0	0	0	1	1	03	—	—	—	—	PW_{11}	PW_{10}	PW_9	PW_8	PW HI	WRITE-ONLY
4	0	0	1	0	0	04	NOISE	⊓⊓	⊿⊿	/\/\	TEST	RING MOD	SYNC	GATE	CONTROL REG	WRITE-ONLY
5	0	0	1	0	1	05	ATK_3	ATK_2	ATK_1	ATK_0	DCY_3	DCY_2	DCY_1	DCY_0	ATTACK/DECAY	WRITE-ONLY
6	0	0	1	1	0	06	STN_3	STN_2	STN_1	STN_0	RLS_3	RLS_2	RLS_1	RLS_0	SUSTAIN/RELEASE	WRITE-ONLY
															Voice 2	
7	0	0	1	1	1	07	F_7	F_6	F_5	F_4	F_3	F_2	F_1	F_0	FREQ LO	WRITE-ONLY
8	0	1	0	0	0	08	F_{15}	F_{14}	F_{13}	F_{12}	F_{11}	F_{10}	F_9	F_8	FREQ HI	WRITE-ONLY
9	0	1	0	0	1	09	PW_7	PW_6	PW_5	PW_4	PW_3	PW_2	PW_1	PW_0	PW LO	WRITE-ONLY
10	0	1	0	1	0	0A	—	—	—	—	PW_{11}	PW_{10}	PW_9	PW_8	PW HI	WRITE-ONLY
11	0	1	0	1	1	0B	NOISE	⊓⊓	⊿⊿	/\/\	TEST	RING MOD	SYNC	GATE	CONTROL REG	WRITE-ONLY
12	0	1	1	0	0	0C	ATK_3	ATK_2	ATK_1	ATK_0	DCY_3	DCY_2	DCY_1	DCY_0	ATTACK/DECAY	WRITE-ONLY
13	0	1	1	0	1	0D	STN_3	STN_2	STN_1	STN_0	RLS_3	RLS_2	RLS_1	RLS_0	SUSTAIN/RELEASE	WRITE-ONLY
															Voice 3	
14	0	1	1	1	0	0E	F_7	F_6	F_5	F_4	F_3	F_2	F_1	F_0	FREQ LO	WRITE-ONLY
15	0	1	1	1	1	0F	F_{15}	F_{14}	F_{13}	F_{12}	F_{11}	F_{10}	F_9	F_8	FREQ HI	WRITE-ONLY
16	1	0	0	0	0	10	PW_7	PW_6	PW_5	PW_4	PW_3	PW_2	PW_1	PW_0	PW LO	WRITE-ONLY
17	1	0	0	0	1	11	—	—	—	—	PW_{11}	PW_{10}	PW_9	PW_8	PW HI	WRITE-ONLY
18	1	0	0	1	0	12	NOISE	⊓⊓	⊿⊿	/\/\	TEST	RING MOD	SYNC	GATE	CONTROL REG	WRITE-ONLY
19	1	0	0	1	1	13	ATK_3	ATK_2	ATK_1	ATK_0	DCY_3	DCY_2	DCY_1	DCY_0	ATTACK/DECAY	WRITE-ONLY
20	1	0	1	0	0	14	STN_3	STN_2	STN_1	STN_0	RLS_3	RLS_2	RLS_1	RLS_0	SUSTAIN/RELEASE	WRITE-ONLY
															Filter	
21	1	0	1	0	1	15	—	—	—	—	—	FC_2	FC_1	FC_0	FC LO	WRITE-ONLY
22	1	0	1	1	0	16	FC_{10}	FC_9	FC_8	FC_7	FC_6	FC_5	FC_4	FC_3	FC HI	WRITE-ONLY
23	1	0	1	1	1	17	RES_3	RES_2	RES_1	RES_0	FILTEX	FILT 3	FILT 2	FILT 1	RES/FILT	WRITE-ONLY
24	1	1	0	0	0	18	3 OFF	HP	BP	LP	VOL_3	VOL_2	VOL_1	VOL_0	MODE/VOL	WRITE-ONLY
															Misc.	
25	1	1	0	0	1	19	PX_7	PX_6	PX_5	PX_4	PX_3	PX_2	PX_1	PX_0	POT X	READ-ONLY
26	1	1	0	1	0	1A	PY_7	PY_6	PY_5	PY_4	PY_3	PY_2	PY_1	PY_0	POT Y	READ-ONLY
27	1	1	0	1	1	1B	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0	OSC_3/RANDOM	READ-ONLY
28	1	1	1	0	0	1C	E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	ENV_3	READ-ONLY

SETTING VOLUME — SAME FOR ALL 3 VOICES		
VOLUME CONTROL	POKE54296	Settings range from O (off) to 15 (loudest)

TO CONTROL THIS SETTING:	POKE THIS NUMBER:	FOLLOWED BY ONE OF THESE NUMBERS (0 to 15 . . . or . . . 0 to 255 depending on range)

TO PLAY A NOTE		C	C#	D	D#	E	F	F#	G	G#	A	A#	B
HIGH FREQUENCY	54273	33	35	37	39	42	44	47	50	53	56	59	63
LOW FREQUENCY	54272	135	134	162	223	62	193	107	60	57	99	190	75

WAVEFORM	POKE	TRIANGLE	SAWTOOTH	PULSE	NOISE
	54276	17	33	65	129

PULSE RATE (Pulse Waveform)		
HI PULSE	54275	A value of 0 to 15 (for Pulse waveform only)
LO PULSE	54274	A value of 0 to 255 (for Pulse waveform only)

ATTACK/DECAY	POKE	ATK4	ATK3	ATK2	ATK1	DEC4	DEC3	DEC2	DEC1
	54277	128	64	32	16	8	4	2	1

SUSTAIN/RELEASE	POKE	SUS4	SUS3	SUS2	SUS1	REL4	REL3	REL2	REL1
	54278	128	64	32	16	8	4	2	1

TO PLAY A NOTE		C	C#	D	D#	E	F	F#	G	G#	A	A#	B
HIGH FREQUENCY	54280	33	35	37	39	42	44	47	50	53	56	59	63
LOW FREQUENCY	54279	135	134	162	223	62	193	107	60	57	99	190	75

WAVEFORM	POKE	TRIANGLE	SAWTOOTH	PULSE	NOISE
	54283	17	33	65	129

PULSE RATE (Pulse Waveform)		
HI PULSE	54282	A value of 0 to 15 (for Pulse waveform only)
LO PULSE	54281	A value of 0 to 255 (for Pulse waveform only)

ATTACK/DECAY	POKE	ATK4	ATK3	ATK2	ATK1	DEC4	DEC3	DEC2	DEC1
	54284	128	64	32	16	8	4	2	1

SUSTAIN/RELEASE	POKE	SUS4	SUS3	SUS2	SUS1	REL4	REL3	REL2	REL1
	54285	128	64	32	16	8	4	2	1

TO PLAY A NOTE		C	C#	D	D#	E	F	F#	G	G#	A	A#	B
HIGH FREQUENCY	54287	33	35	37	39	42	44	47	50	53	56	59	63
LOW FREQUENCY	54286	135	134	162	223	62	193	107	60	57	99	190	75

WAVEFORM	POKE	TRIANGLE	SAWTOOTH	PULSE	NOISE
	54290	17	33	65	129

PULSE RATE (Pulse Waveform)		
HI PULSE	54289	A value of 0 to 15 (for Pulse waveform only)
LO PULSE	54288	A value of 0 to 255 (for Pulse waveform only)

ATTACK/DECAY	POKE	ATK4	ATK3	ATK2	ATK1	DEC4	DEC3	DEC2	DEC1
	54291	128	64	32	16	8	4	2	1

SUSTAIN/RELEASE	POKE	SUS4	SUS3	SUS2	SUS1	REL4	REL3	REL2	REL1
	54292	128	64	32	16	8	4	2	1

tabel geluidswaarden

Instelwaarden voor diverse instrumenten

Instrument	Golfvorm	Attack	Sustain	Pulstijd
Piano	Puls	9	0	Hi-0 Lo-255
Fluit	Driehoek	96	0	nvt
Clavecimbel	Zaagtand	9	0	nvt
Xylofoon	Driehoek	9	0	nvt
Orgel	Driehoek	0	240	nvt
Accordeon	Driehoek	102	0	nvt
Trompet	Zaagtand	96	0	nvt

NOOT: De instellingen van aanzweltijd/verval en nagalmvolume/uitsterfsnelheid moeten worden ingesteld v o o r het instellen van de golfvorm!

INDEX

Trefwoord — Pagina's

A

Trefwoord	Pagina's
Aanhaalteken	28
Aansluit gegevens	145-147
Aansluitingen	5-23
Aansluitingen (tv en monitor)	5-11
Aansluitingen (zijkant)	5-6
Accessoires	2 18-20 103-108
Afkorting (BASIC commando)	134-135
Aftrekken	29 121
AND operator	121
Animaties	43-66
ASC functie	132 139-141

B

Trefwoord	Pagina's
BASIC (afkortingen)	134-135
BASIC (commando's)	122-123
BASIC (instructies)	123-125
BASIC (numerieke functies)	125-127
BASIC (operators)	121
BASIC (overige functies)	129
BASIC (string functies)	128
BASIC (variabelen)	120-121
Beginnen is moeilijk	21-32
Berekeningen	28-32
Bestanden of files	27 109-119
Binair rekenen	64-66
Bit	64-66
Byte	64-66

C

Trefwoord	Pagina's
Cassette aansluiting	14 109-119
Cassetterecorder	14 109-119
CHR$ functie	51-52 120 132 139-141 151
CLOSE instructie	124
CLR instructie	124
CLR/HOME toets	23
Commando's (BASIC)	122-124
Commodore toets	23
CONT commando	122
CONTROL toets	16 23
Corrigeren	35
Cosinus functie	131

Cursor 16
Cursor toetsen 16 23

D

Data (LOAD en SAVE) 24-27
DATA regels 89-91 124
Datasette recorder (zie cassette recorder)
DEF instructie 125
DEL toets 23
Delen 29 30 31 121
DIM instructie 125 126
Dimensioneren 125 126

E

END instructie 125
EXP functie 131

F

Files of bestanden 26 109-119
FOR instructie 125
Formules (rekenkundige) 29 30-31 121 126 144
Fout meldingen 28-29 152-153
FRE functie 133
Functie's omschrijven (zie DEF instructie)
Functies 130-133
Functie toetsen 47

G

Game paddles 4-5 145
Geheugen indeling 62-65
Geheugen uitbreiden 5-5 146-147
Geïndiceerde variabelen 93-99 120-121
Gelijk aan teken 29 30-33 122
Geluidseffecten 86-87
GET instructie 46-48 126
GET instructie 126
GOSUB instructie 126
GOTO of GO TO instructie 35-34 126
Grafische symbolen (zie grafische toetsen)
Grafische toetsen 24 54-55 58 136-141
Grote en kleine letters 22
Groter dan teken 121

H

Haakjes 32
Hyperbolische functies 144

I

I/O aansluitingen 145-148
I/O poorten 4-13 145-148
IEEE-488 interface 4-13 145
IF...THEN instructie 38-39 126-127
INPUT instructie 45-46 127
INPUT # 127
INST toets 23
INT functie 131
Integere variabelen 120

J

Joysticks 4-5 145

K

Kleine letters 22-24
Kleur afstelling 16-17
Kleur codes 56
Kleur geheugen 60 143
Kleur PEEK en POKE 58-59
Kleur scherm en rand 58-59 142
Kleur toetsen 54-55
Klok (intern) 121

L

Laden van programma's 24-27
LEFT$ functie 132
LEN (lengte) functie 132
LET instructie 127
LIST commando 35-35 123
Literatuuroverzicht 156-158
LOAD commando 123
LOG (logarithme) functie 131
Loop of lus 39-40 43-45

M

Machtsverheffen 30-31 121
Matrix 93-100
MID$ functie 132
Minder dan 121
Modulator (HF) 6-12
Muziek 75-87

N

Naam (van programma) 24-27
Naam (van variabele) 35-38

NEW commando 123
NEXT instructie 127-128
Niet gelijk aan 29 30-31 122
NOT operator 122
Numerieke variabelen 35-36

O

ON instructie 128
Onderkasten 22-25
OPEN instructie 128
Operator (logische-) 121
Operator (relationele-) 121
Operator (wiskundige-) 121
Operators (rekenkundige) 29 30-31 121 126 144
Opmerkingen (in programma) 130

P

Parenthesen 32
PEEK functie 58-59
POKE instructie 58-59
Poorten (voor I/O) 6-12 145-148
POS functie 133
PRINT instructie 29-32 129
PRINT 129
Programma's (LOAD en SAVE) 24-27
Programma's (regelno's van-) 34-35
Programma's (wijzigen van-) 20-35

R

Randapparaten 4-15 18-20
Random getallen 48-52
READ instructie 129
REM instructie 130
RESTORE instructie 130
RESTORE toets 23-24
RETURN instructie 130
RETURN toets 23-25
RIGHT$ functie 132
RND functie 48-53 131
RUN commando 123
RUN/STOP 23

S

SAVE commando 27 123
SAVE van programma's 27
Scherm (geheugenindeling) 59-60 142
Schrijven naar tape 109

SGN functie 132
SHIFT toets 22-24
SIN (sinus) functie 132
Sleutelwoorden (zie BASIC woorden)
SPC (spaties) functie 133
Spellen (software) 103-108
Spellen (in- en uitgangen) 7-15 145-148
Sprites 64-74
Sprites maken 64-74
SQR (worteltrekken) 132
Stem 76-84 163-164
Stop commando 130
Stop toets 23-24
STR$ functle 133
String (tekst) variabelen 37-38 120-121
Syntax error 28
SYS instructie 130

T

TAB functie 133
TAN (tangens) functie 132
Televisie (verbindingen) 5-12
TI variabele 121
TI$ variabele 121
Tijd 121
Toetsenbord 22-24
Tussenvoegen 23

U

Uitgangen 145-148
USR functie 132

V

VAL functie 132
Variabelen (dimensie) 95-103 121
Variabelen (integere-) 95-103 120
Variabelen (matrix-) 95-103 120
Variabelen (numerieke-) 95-103 120
Variabelen (string- of tekst-) 95-103 120
Variabelen (vlottende komma) 95-103 120
Vergelijkingen 122
VERIFY commando 124
Vermenigvuldigen 29 121
Vertragen (zie for...next)
Videosignaal 5-15
Vraagteken 45

W

Wachtlus	58 61
WAIT commando	130
Wijzigen (programma's)	23 36
Willekeurig (getal)	48-53
Wiskundige formules	29-31
Wiskundige functies	144
Wiskundige symbolen	29-31 38 122
Worteltrekken (SQR)	132

Z

Zakelijke programma's	103-108

Commodore hoopt dat u veel plezier hebt beleefd aan dit gebruikershandboek. U moet het, ondanks het feit dat er allerlei tips en informatie in staat, niet beschouwen als het handboek voor de echte programmeur, die alle aspecten in detail wil kennen en gebruiken.

Voor programmeurs en gevorderden bevelen wij de aanschaf aan van het boek "Commodore 64 Programmers Reference Guide". Dit boek is via uw leverancier te verkrijgen.